Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National Polytechnique de Toulouse (INP Toulouse)

Présentée et soutenue par :
Juliette HEINTZ
Le mardi 23 octobre 2012

Titre :

Systemic approach and decision process for sustainability in chemical engineering: Application to computer aided product design

---

EDSYS : Génie Industriel 4200046

Unité de recherche :
Laboratoire de Génie Chimique, UMR 5503 - Département Procédés et Systèmes Industriels

Directeur(s) de Thèse :

Vincent GERBAUD, Jean-Pierre BELAUD

Rapporteurs :

Kyle V. CAMARDA, Aline CAUVIN

Autre(s) membre(s) du jury :

Jean-Marc Le LANN, Xavier JOULIA, Emmanuel CAILLAUD,
Philippe MARION, Jean-Charles de HEMPTINNE

Abstract

In a context where environmental issues are increasingly taken into account, the chemical related industry faces situations imposing a chemical product substitution. Computer aided molecular design methods, which consist in finding molecules satisfying a set of constraints, are well adapted to these situations.

Using a systemic analysis of the needs and uses linked to this context, we develop a computer aided product design tool implementing a genetic algorithm. It is able to explore a wider solution space thanks to a flexible molecular framework. Besides, by allowing a very flexible setting of the problem to be solved, it enables the search of molecules sourced from renewable resources.

Based on concepts from system and enterprise engineering, we formalize a decision making process dedicated to the product substitution in an industrial context. This multi-criteria decision process includes the phases of the requirements definition, of the generation of alternative solutions, of the selection of the best alternative and of the product application. It uses a model driven approach and decision making techniques that guaranty an operational alignment in addition to the strategic alignment across the chemical enterprise.

Through a case study, we expose how the combination of our computer aided product design tool and our decision making process enables an environmentally compliant approach of product substitution which is both efficient and in adequacy with enterprise context.

Key Words

Process system engineering – Computer aided product design – Computer aided molecular design – Industrial systems – Enterprise engineering – Decision making – Sustainability – Biosourced solvents.

Résumé

Titre : Approche systémique et processus décisionnel pour le développement durable en génie des procédés: Application à la substitution de produits par formulation inverse

Dans un contexte de prise en compte croissante des enjeux environnementaux, l'industrie de la chimie et des procédés se retrouve confrontée à des problématiques de substitution de molécules. Les méthodes de formulation inverse, qui consistent en la recherche assistée par ordinateur de molécules satisfaisant un ensemble de contraintes, répondent de manière efficace à ces problématiques.

A partir de l'analyse systémique des usages et fonctionnalités nécessaires dans ce contexte, nous développons un outil logiciel de formulation inverse mettant en œuvre un algorithme génétique. Celui-ci est capable d'explorer un espace de solutions plus vaste en considérant les mélanges et non les molécules seules. Par ailleurs, il propose une définition des problèmes très flexible qui permet la recherche efficiente de molécules issues de filières renouvelables.

En s'appuyant sur l'ingénierie système et l'ingénierie d'entreprise, nous proposons un processus formel de prise de décision pour la substitution de produit dans un contexte industriel. Ce processus de décision multi-critères englobe les phases de définition des exigences, de génération de solutions alternatives, de sélection de la meilleure alternative et de mise en œuvre du produit. Il utilise une approche dirigée par les modèles et des techniques de prises de décision qui garantissent un alignement opérationnel en complément de l'alignement stratégique.

A travers un cas d'étude, nous montrons comment l'utilisation conjointe de notre outil de recherche par formulation inverse et de notre processus de décision permet une démarche environnementale de substitution de produit à la fois efficiente et conforme à la réalité de l'entreprise.

Mots clés

Génie des procédés – Formulation inverse de molécules assistée par ordinateur – Conception de produits assistée par ordinateur – Systèmes Industriels – Ingénierie d'entreprise – Prise de décision – Développement durable – solvants biosourcés.

Remerciements

Ce travail de thèse a été réalisé sous la direction de Messieurs Vincent Gerbaud, Chargé de Recherche CNRS, et Jean-Pierre Belaud, Enseignant chercheur à L'Ensiacet. Je tiens à les remercier pour m'avoir fait confiance, pour m'avoir soutenue et pour avoir été disponibles tout au long de ces trois années.

Je remercie mes rapporteurs, Monsieur Kyle Camarda, Associate Professor au département Chemical and Petroleum Engineering de l'Université du Kansas, et Madame Aline Cauvin, Maitre de Conférences au Laboratoire des Sciences de l'Information et des Systèmes de l'Université Aix-Marseille, pour l'intérêt qu'ils ont porté à mes travaux. Je remercie également les membres du Jury : Monsieur Xavier Joulia, Monsieur Emanuel Caillaud, Monsieur Jean-Marc Le Lann, Monsieur Philippe Marion et Monsieur Jean-Charles de Hemptinne qui m'ont fait l'honneur de discuter de mes travaux et qui m'ont permis d'entrevoir des perspectives dans leurs domaines respectifs.

Je remercie également tous les membres du projet InBioSynSolv et plus particulièrement Moises Teles dos Santos et Nishant Pandya avec qui j'ai travaillé en étroite collaboration lors du développement logiciel. Je tiens aussi à remercier Manon Bergez-Lacoste, Laurianne Moity et Anthony Naralingom qui ont été les premiers utilisateurs du logiciel et qui ont fortement contribué à son amélioration grâce à leurs remarques toujours pertinentes.

Je remercie également Monsieur Rafiqul Gani, Professeur à CAPEC DTU, précurseur du CAMD, dont les travaux ont été une grande source d'inspiration notamment ceux de son étudiante Elisa Conte.

Je souhaite également exprimer ma reconnaissance à tout le département PSI et plus particulièrement aux doctorants de la rue 3. Grâce à la bonne humeur et l'entraide qui règnent au sein de ce groupe, cela a été un réel plaisir de venir travailler tous les jours. Je souhaiterai aussi remercier Marie Roland et Iréa Touche pour les longues pauses et discussions qui m'ont remonté le moral.

Je termine en remerciant toute ma petite famille pour son soutien affectif et pour ses encouragements continus, qui m'ont aidée à maintenir le cap et à surmonter les moments difficiles.

Table of contents

A. INTRODUCTION 1

1. CONTEXT AND OBJECTIVES 3

1.1 INDUSTRIAL CONTEXT 4

1.2 SCIENTIFIC OBJECTIVES 5

1.3 EXPECTED BENEFITS 6

1.4 RESEARCH FRAME 7

1.5 MANUSCRIPT PRESENTATION 8

B. COMPUTER AIDED PRODUCT DESIGN IN THE CHEMICAL FIELD 11

2. STATE OF THE ART 13

2.1 FUNDAMENTAL PRINCIPLES OF COMPUTER AIDED MOLECULAR DESIGN 14

2.2 MOLECULAR REPRESENTATION MODELS 16

2.3 NUMERICAL RESOLUTION METHODS 24

2.4 PROPERTIES AND MODELS 36

2.5 MIXTURE PERFORMANCE EVALUATION 40

2.6 COMPUTER AIDED MIXTURE DESIGN 46

2.7 CONCLUSION 48

3. PROPOSITION OF A COMPUTER AIDED PRODUCT DESIGN METHOD 51

3.1 FROM MOLECULE AND MIXTURE DESIGN TO PRODUCT DESIGN 52

3.2 SEARCH ALGORITHM 57

3.3 MOLECULE MANAGEMENT 60

3.4 PERFORMANCE CALCULATION 66

3.5 CONCLUSION 68

4. DEVELOPMENT OF THE IBSS TOOL 69

4.1 SOFTWARE DEVELOPMENT PROCESS 70

4.2 System views 72
4.3 The functional view 73
4.4 The architectural view 78
4.5 The structural view 80
4.6 The Behavioral view 93
4.7 Deployment 100
4.8 Validation 101
4.9 Conclusion 104

C. A DECISION MAKING PROCESS FOR CHEMICAL PRODUCT SUBSTITUTION 107

5. STATE OF THE ART 109
5.1 Enterprise modeling 111
5.2 Model driven engineering 113
5.3 Decision Making Process 115
5.4 Requirements Management 122
5.5 Modeling languages 125
5.6 Conclusion 128

6. PROPOSITION OF A DECISION MAKING PROCESS FOR CHEMICAL PRODUCT DESIGN 129
6.1 Decision making process overview 130
6.2 Intelligence phase 131
6.3 Design phase 138
6.4 Choice Phase 141
6.5 General picture of our proposal for a product substitution decision process 146
6.6 Conclusion 147

7. INDUSTRY RELATED CASE STUDY 149
7.1 Context 150
7.2 Stimulus 150
7.3 Intelligence phase 151

7.4 DESIGN PHASE 163

7.5 CHOICE PHASE 166

7.6 CONCLUSION 170

D. EPILOGUE 173

8. CONCLUSION AND PERSPECTIVES 175

8.1 CONCLUSION 176

8.2 SYNTHESIS OF SCIENTIFIC CONTRIBUTIONS 180

8.3 LIMITATIONS AND PERSPECTIVES 181

9. BIBLIOGRAPHY 185

10. APPENDIXES 193

10.1 BASIC FUNCTIONAL GROUP CODING 194

10.2 LIST OF CALCULATION MODELS 199

10.3 UML DIAGRAMS 203

10.4 GRAPH CONSTRUCTION METHOD 205

10.5 COMPOSITION HANDLING 216

10.6 IBSS XML INPUT FILE 219

10.7 IBSS OUTPUT FILE EXTRACT 228

10.8 DELPHI QUESTIONNAIRE 247

10.9 ACRONYMS 255

Figures

FIGURE 1: POSITIONING OF OUR WORK 8

FIGURE 2: PRESENTATION OF A PART 9

FIGURE 3: THE TRIAL AND ERROR AND THE CAMD METHODOLOGY 15

FIGURE 4: CAMD METHODS 16

FIGURE 5: EXAMPLE OF A STRING REPRESENTATION AS USED BY PATKAR AND VENKATASUBRAMANIAN (2002) 17

FIGURE 6: EXAMPLES OF COLLECTIONS OF GROUPS AS USED IN GANI'S METHOD 18

FIGURE 7: BINARY REPRESENTATION OF 1,1,1-TRICHRO-2-2-DIFLUOROETHANE (FROM CHURI AND ACHENIE, 1996) 19

FIGURE 8: EXAMPLES OF "CLASSIC" GRAPH REPRESENTATIONS AS USED BY RAMAN AND MARANAS (1998) 20

FIGURE 9: LACK OF PRECISION OF THE GRAPH REPRESENTATION MODEL 20

FIGURE 10: EXAMPLE OF THE MOLECULAR REPRESENTATION USED BY LIN ET AL. (2005) 21

FIGURE 11: EXAMPLE OF THE MATRIX REPRESENTATION OF OURIQUE AND SILVA TELLES (1998) 21

FIGURE 12: SIGNATURE TRANSLATION EXAMPLE (FROM WEIS AND VISCO, 2010) 22

FIGURE 13: HARPER ET AL. MULTI-LEVELED MOLECULAR REPRESENTATION ALGORITHM 26

FIGURE 14: MOLECULAR GRAPH ADAPTATION TO PROPERTY ESTIMATION METHODS (FROM FIGURE 3 IN (KORICHI ET AL., 2008)) 27

FIGURE 15: GENERAL STEPS OF A SIMULATED ANNEALING ALGORITHM 32

FIGURE 16: GENERAL STEPS OF A TABU SEARCH 34

FIGURE 17: GENERAL STEPS OF A GENETIC ALGORITHM 35

FIGURE 18: PERFORMANCE CALCULATED WITH A SCALED DEVIATION 40

FIGURE 19: PERFORMANCE CALCULATED WITH A GAUSSIAN FUNCTION 41

FIGURE 20: PERFORMANCE CALCULATED FOR A MAXIMIZATION WITH A SIGMOID FUNCTION 42

FIGURE 21: PERFORMANCE CALCULATED FOR A MINIMIZATION WITH A SIGMOID FUNCTION 43

FIGURE 22: PERFORMANCE CALCULATED FOR A MAXIMIZATION WITH A DESIRABILITY FUNCTION 44

FIGURE 23: PERFORMANCE CALCULATED FOR A TARGET VALUE OBJECTIVE AND A PROPERTY CONSTRAINT WITH DESIRABILITY FUNCTIONS 44

FIGURE 24: OPTIMIZATION VARIABLES OF MIXTURE DESIGN 52

FIGURE 25: MOLECULAR SPECIFICATION POSSIBILITIES 53

FIGURE 26: EXAMPLE OF THE DEFINITION OF THE TARGET VALUE OF A REAL PROPERTY 55

FIGURE 27: PROPERTY MANAGEMENT 56
FIGURE 28: BASIC PRINCIPLE OF THE GENETIC ALGORITHM (BPMN DIAGRAM) 59
FIGURE 29: BASIC PRINCIPLE OF THE MULTI-LEVELED GENETIC ALGORITHM (BPMN DIAGRAM) 60
FIGURE 30: MOLECULAR GRAPH REPRESENTATION OF A MOLECULE 61
FIGURE 31: FRAGMENT REPRESENTATION OF A MOLECULE 62
FIGURE 32: SIMPLE AND COMPLEX GROUP REPRESENTATION OF A MOLECULE 62
FIGURE 33: MUTATION OPERATOR EXAMPLE 63
FIGURE 34: CROSSOVER OPERATOR EXAMPLE 63
FIGURE 35: INSERTION OPERATOR EXAMPLE 64
FIGURE 36: DELETION OPERATION EXAMPLE 65
FIGURE 37: SUBSTITUTION OPERATOR EXAMPLE 65
FIGURE 38: THE SYSTEM VIEWS 72
FIGURE 39: THE DIFFERENT TYPES OF USERS 73
FIGURE 40: "LAUNCH A SEARCH" USE CASE DIAGRAM 74
FIGURE 41: "CREATE A CAPD PROBLEM" AND "CREATE AN OBJECTIVE FUNCTION" USE CASE DIAGRAM 75
FIGURE 42: "CREATE A PROPERTY" AND "CREATE A MODEL" USE CASE DIAGRAM 76
FIGURE 43: "EVALUATE MIXTURES" USE CASE DIAGRAM 77
FIGURE 44: COMPONENT DIAGRAM 78
FIGURE 45: INTERFACE OF PROPERTY CALCULATION SERVICE (MOLPROPESTIM.DLL) 79
FIGURE 46: PACKAGE DIAGRAM 80
FIGURE 47: THE MMI PLUG PACKAGE 81
FIGURE 48: ROOT OF THE PROBLEM PACKAGE 82
FIGURE 49: STRUCTURE OF THE OBJECTIVE FUNCTION DATA 83
FIGURE 50: STRUCTURE OF THE TARGET DATA 84
FIGURE 51: STRUCTURE OF MIXTURE DATA 84
FIGURE 52: STRUCTURE OF THE COMPOSITION DATA OF AN ELEMENT IN THE MIXTURE 85
FIGURE 53: STRUCTURE OF MOLECULE DATA 86
FIGURE 54: ILLUSTRATION OF A FREE MOLECULE 86
FIGURE 55: STRUCTURE OF MOLECULAR FRAGMENT DATA 87
FIGURE 56: STRUCTURE OF THE OPERATING CONDITION DATA 87

FIGURE 57: ROOT OF THE RESOLUTION PACKAGE 88
FIGURE 58: STRUCTURE OF A MIXTURE POPULATION 89
FIGURE 59: GRAPHCONSTRUCTOR AND GRAPHMODIFIER CLASS 89
FIGURE 60: CLASS DIAGRAM REPRESENTING THE DIFFERENT TYPES OF FUNCTIONAL GROUPS 90
FIGURE 61: INFORMATION ASSOCIATED TO A COMPLEX GROUP 92
FIGURE 62: EXAMPLE OF A COMPLEX GRAPH AND AN EXPANDED GRAPH 92
FIGURE 63: BPMN DIAGRAM GIVING A GLOBAL OVERVIEW OF THE SYSTEM BEHAVIOR 93
FIGURE 64: MACRO-PROCESS OF THE SPECIFICATION OF A PROBLEM DATA 94
FIGURE 65: SIMPLIFIED PROCESS OF A MIXTURE CREATION 95
FIGURE 66: ACTIVITY DIAGRAM OF THE CREATION OF A FREEFRAGMENT OBJECT 97
FIGURE 67: SIMPLIFIED PROCESS OF A MIXTURE MODIFICATION 99
FIGURE 68: DEPLOYMENT DIAGRAM OF THE RESTRICTED DEPLOYMENT 100
FIGURE 69: DEPLOYMENT DIAGRAM OF THE STANDALONE DEPLOYMENT 101
FIGURE 70: DEPLOYMENT DIAGRAM OF THE DISTRIBUTED DEPLOYMENT 101
FIGURE 71: CHEMICAL SUPPLY CHAIN INSPIRED BY (MARQUARDT ET AL., 2000) 111
FIGURE 72: THE FOUR VIEWS OF THE ENTERPRISE FROM (TOUZI, 2008) 113
FIGURE 73: THE REAL SYSTEM, THE MODEL AND THE METAMODEL 114
FIGURE 74: OUTLINE OF THE DECISION MAKING PROCESS (FROM VALLIN AND VANDERPOOTEN, 2000) 116
FIGURE 75: CAUVIN'S DECISION PROCESS (FROM CAUVIN, 2005) 117
FIGURE 76: HOWARD'S DECISION ANALYSIS PROCEDURE (HOWARD, 1966) 118
FIGURE 77: PLANNING AND CONTROL PROCESSES IN ORGANIZATION (ANTHONY, 1965) 121
FIGURE 78: ILOG JRULES 124
FIGURE 79: SVBR STRUCTURED ENGLISH EXAMPLE 127
FIGURE 80: RULESPEAK EXAMPLE (FROM WWW.BRCOMMUNITY.COM) 127
FIGURE 81: OVERVIEW OF OUR DECISION MAKING PROCESS 130
FIGURE 82: BUSINESS PLAYERS IN THE INTELLIGENCE PHASE 132
FIGURE 83: A GLOBAL MULTI-LEVEL AND MULTI-LAYER APPROACH FOR THE INTELLIGENCE PHASE FROM HEINTZ ET AL, (2012) 133
FIGURE 84: A SIMPLIFIED MULTI-LEVEL AND MULTI-LAYER APPROACH FOR THE INTELLIGENCE PHASE 135
FIGURE 85: AN UML MODEL OF REQUIREMENT 135
FIGURE 86: THE DESIGN PHASE 139

FIGURE 87: THE CHOICE PHASE 141

FIGURE 88: GENERAL PICTURE OF OUR DECISION MAKING PROCESS 146

FIGURE 89: REQUIREMENTS OF THE STRATEGIC DECISION MAKING LEVEL 152

FIGURE 90: FORMALISM OF THE REQUIREMENTS TREE 153

FIGURE 91: REQUIREMENTS AT THE TACTICAL DECISION MAKING LEVEL 154

FIGURE 92: CLASS DIAGRAM USED FOR THE OCL DESCRIPTION OF THE CONSTRAINTS ON THE PROPERTIES 154

FIGURE 93: PARTIAL VIEW OF REQUIREMENTS ON THE PROPERTY TARGETS 158

FIGURE 94: CLASS DIAGRAM USED FOR THE OCL DESCRIPTION OF THE CONSTRAINTS ON THE MOLECULAR STRUCTURES 159

FIGURE 95 PARTIAL VIEW OF REQUIREMENTS ON THE MOLECULAR STRUCTURES 161

FIGURE 96: DISTRIBUTION OF THE MODIFICATION PROBABILITIES AMONG THE ELEMENT DEFINING A MIXTURE 164

FIGURE 97: BEST BIOMASS BASED MOLECULE AND INFLUENCE OF ITS FRACTION ON RED 169

FIGURE 98: CALCULATION OF THE REAL CONSTRAINTS 217

FIGURE 99: RANDOM CHOICE OF A COMPOSITION VALUE 217

FIGURE 100: RANDOM CHOICE OF A COMPOSITION VALUE THAT FINALIZE THE DETERMINATION OF THE COMPOSITION VECTOR 218

FIGURE 101: EXAMPLE OF A COMPOSITION MODIFICATION 219

Tables

TABLE 1: SUMMARY TABLE OF THE COMMONLY USED MOLECULAR REPRESENTATION IN CAMD 23

TABLE 2: CLASSIFICATION OF THE GROUPS BY (GANI ET AL., 1991) 24

TABLE 3: SUMMARY TABLE OF THE SEARCH METHODS USED IN CAMD 36

TABLE 4: PERFORMANCE FUNCTIONS 66

TABLE 5: REAL PROPERTIES AVAILABLE IN OUR APPLICATION 77

TABLE 6: TEN BEST ACYCLIC MOLECULES MATCHING A 358.15K BOILING POINT TARGET 102

TABLE 7: TEN BEST MOLECULES MATCHING A 358.15K BOILING POINT AND 100K MELTING POINT TARGET 103

TABLE 8: TYPES OF DECISIONS AND ASSOCIATED SITUATIONS (FROM CAUVIN, 2005) 115

TABLE 9: ANSOFF'S DECISION MAKING MODEL (FROM ANSOFF, 1965) 120

TABLE 10: PARAMETERS OF THE PROPERTY CONSTRAINTS COMING FROM PRIORITY REQUIREMENTS 156

TABLE 11: PARAMETERS OF THE PROPERTY CONSTRAINTS COMING FROM NON-PRIORITY REQUIREMENTS 157

TABLE 12: SUMMARY TABLE OF THE PROPERTIES, THEIR TARGET, THEIR MODEL AND THE ASSOCIATED PARAMETERS 162

FIRST PART

A. INTRODUCTION

1. Context and Objectives …….. 3

ABSTRACT

This part outlines the scientific context in which our work is performed, our objectives and the expected benefits of our work. Then the research frame of our work is described and all the themes that are addressed are presented. Finally the structure of the document is detailed and the different parts are described.

CHAPTER 1

Context and Objectives

1.1 INDUSTRIAL CONTEXT

1.1.1 A need for eco-friendly chemical products

In the last decades, the necessity to take into account environmental issues has grown into a global awareness in our society. Companies are now pushed toward sustainability by regulations and by customer demand. A commonly used definition of sustainable development is the definition given in the Brundtland commission in 1987. It defines sustainable development as a "development that meets the needs of the present without compromising the ability of future generations to meet their own needs". The chemical industry, which has for purpose to convert raw materials into a very wide range of products, is particularly concerned by this evolution. Indeed, this industry is largely associated to the ecological impact of chemical product waste and to the consumption of non-renewable natural resources. The REACH regulation[1] or the VOC directives[2] goes this way by imposing strict constraints on the chemical products. Those constraints are forcing chemical companies to give up some of their products or molecules. They then have to find replacement products respecting environmental constraints as well as their production processes and their business strategy.

This represents a real challenge for chemical engineering. Indeed, the substitution methods traditionally used are “trial and error” ones. In these methods, an engineer specialized in chemical synthesis has for mission to find a molecule or product that has specific characteristics on given properties. From this given set of properties, the engineer uses his knowledge and his experiences to select the molecular structures which are the most appropriate according to him. Those structures are then synthesized and their properties are evaluated. For having a chance to find a replacement product, this process must be repeated several times, but a satisfactory outcome is never guaranteed.

With the increasing need to replace products, and with substitutions becoming more and more complex due to the number of constraints to take into account, the chemical industry is repeatedly confronted to challenging problems. Traditional “trial and error” methods must be replaced by more efficient ones, like Computer Aided Molecular Design (CAMD) that can handle simultaneously functional, economical, health, safety and life cycle constraints.

---

[1] http://ec.europa.eu/enterprise/sectors/chemicals/reach/index_en.htm

[2] The VOC Solvents Emissions Directive (Directive 1999/13/EC) amended through article 13 of the Paints Directive (Directive 2004/42/EC)

However, even CAMD methods are becoming overwhelmed. Indeed, with the multiplication of the constraints, it becomes difficult to find molecules satisfying all constraints. A solution is to consider mixtures instead of single molecules.

CAMD methods are also limited by another fact: they request a large domain of expertise in chemical engineering and in chemistry. They are hence not adapted to be used in large scale industries were business strategy and policy are to be taken into account. Expertise in industrial engineering is then also needed.

1.1.2 *The ANR project InBioSynSolv*

Taking note of this situation, the ANR project InBioSynSolv[3] coordinated by Vincent Gerbaud[4] aims first at developing a "virtual laboratory" dedicated to the replacement of solvents, by means of a Computer Aided Product Design (CAPD) software prototype. This project involves several partners. The responsibility of our laboratory is to implement the CAPD tool. The Rhodia group[5] provides the general specifications on the products to be found. The "Laboratoire de Chimie Agroindustrielle" (LCA)[6] and the "Laboratoire de Chimie Organique et Macromoléculaire" (LCOM)[7] find synthons coming from renewable source destined to be used as basis in the product to be found. They are specifically in charge of the synthesis and the testing of the product found by our tool.

This project has initiated the development of a CAPD tool presented in this thesis, but my work goes beyond the scope of this project by proposing a systemic approach and decision process for chemical engineering dedicated to product design.

1.2 SCIENTIFIC OBJECTIVES

There are two main objectives to my PhD work. The first consists in developing an innovative and complete CAPD method and the associate tool. The second is to initiate from a more global reflection on the product substitution process. Both gave rise summarized hereafter.

---

[3] Programme ANR 2008: ANR-09-CP2D-08, Chimie et Procédés pour le développement durable

[4] http://lgc.inp-toulouse.fr/

[5] http://www.rhodia.com/en

[6] http://lca.ensiacet.fr/

[7] http://www.ensc-lille.fr/art96-6-146-laboratoire-de-chimie-moleculaire-et-formulation.html

1.2.1 A innovative CAPD method

The CAPD method we propose has several innovative aspects. Indeed, it goes further than CAMD method as it is a chemical product design method able to find the most suitable mixture by optimizing molecular structures, composition and operating conditions. It also offers the possibility to fix parts of the molecular structures in order to focus the search on synthons coming from renewable sources.

1.2.2 A formalized decision process for product substitution

Having access to such a tool is however not sufficient to tackle the complexity of a chemical product substitution. The task of specifying the requirements is still highly complex in particular in an industrial context where many players are involved. Also, the final choice of the replacement product among the set of results proposed by the tool remains critical.

For those reasons, we have decided to develop a formalized decision making process dedicated to chemical product substitution. This process uses concepts such as model driven engineering, enterprise modeling and decision making methods. It aims at easing the substitution process and ensuring that the product chosen for the substitution is compliant not only with the environmental constraints, but also with the enterprise production processes and business strategy.

1.3 EXPECTED BENEFITS

This innovative approach has the potential to bring significant value to the chemical industry, specifically when it comes to the design of new chemical products. We can identify three main fields of value creation.

- Our proposition can save R&D cost with a search algorithm selecting within few hours of computing the best alternatives among many. The traditional “trial and error” methods would utilize significant human resources, mobilize expensive laboratory assets and consume expensive chemicals.
- Time-to-market is the second source of value that our process can optimize. Within a few days or weeks, a good chemical product alternative can be recommended for implementation in the production process cutting short the long traditional process by months. Therefore fast reaction time can be achieved, providing a leading edge to the

competition and/or enabling continuity of production despite changing regulations and market demand.

- Innovation is the third value proposition of our approach. Since the search algorithm explores randomly the space of solutions, it may explore opportunities that have not been explored before and would not be explored with traditional method because of lacking time and money. Eventually the algorithm will discover solutions that are genuinely innovative and create valuable intellectual property.
- Finally, our decision process can lead to better knowledge management of the process, enabling a faster response to a future similar problem.

1.4 RESEARCH FRAME

In this manuscript, we propose a complete frame which will facilitate the response to situations where the replacement of a chemical product is needed. This had lead us to formalize a decision making process specific to product substitution in an industrial context in which the alignment on the requirements coming from different enterprise levels and expertise domains was ensured, and to develop a method and a software tool of Computer Aided Product Design. Our objective is to grasp the problem in all its complexity and thus consists in a systemic approach as defined by Le Moigne (1994). This work follows the work of two other PhD students from our laboratory:

- Mourad Korichi (2010) who has initiated the works on Computer Aided Molecular Design in our laboratory.
- Jean Stéphane Ulmer (2011) whose work on alignment was a source of inspiration and has oriented our work on themes of the French work group EasyDim[8] which considers the model driven enterprise engineering and information systems.

Our work is at the crossing of several disciplines as presented on Figure 1.

[8] http://www.easy-dim.org/

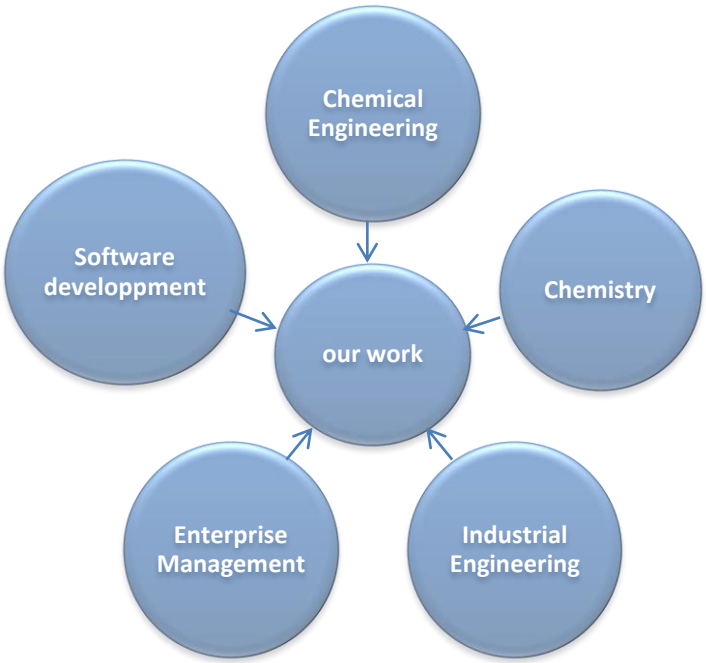


Figure 1: positioning of our work

Concerning Chemical Engineering, our work is more precisely centered on Process System Engineering. In this latter discipline, we focused on Computer Aided Molecular Design methods and associated optimization techniques. Chemistry is also an important part of our work, be it for the formulation of molecules or for determining the needs to which we must adapt. Confronted to our perception of the enterprise as a complex system, and in adequacy with Industrial Engineering, we make efforts to define a systemic approach and we find relevant to use the main concepts of Requirement Engineering and Model Driven Engineering. We also address the themes of Enterprise Management, and in particular Decision Analysis, Business Rule Management and Enterprise and Information System Engineering. We finally relied on Information Software Technologies for the development of our software tool.

All those themes are presented and developed through this manuscript.

1.5 MANUSCRIPT PRESENTATION

Regarding the structure, the thesis is organized in four parts containing 10 chapters. To ease the reading each parts begins with a table of contents (Figure 2-a) and an abstract (Figure 2-b).

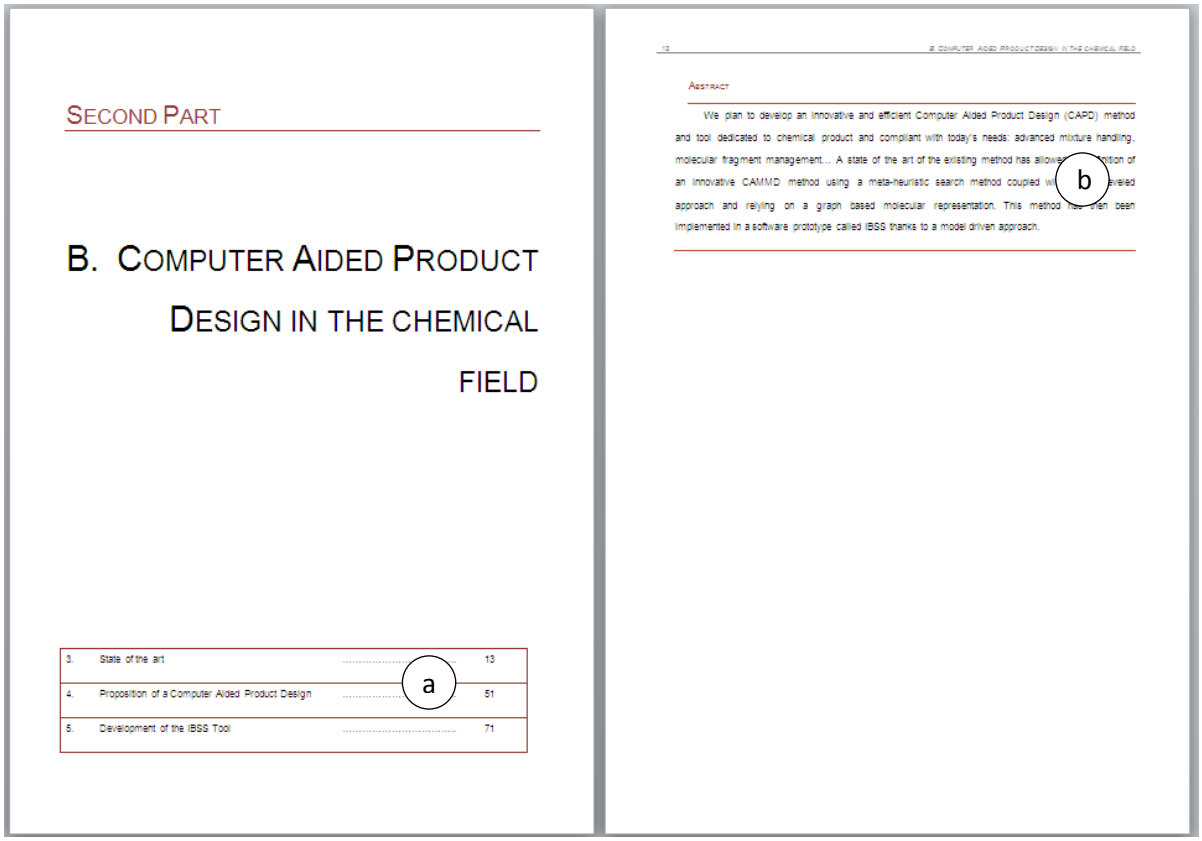


Figure 2: Presentation of a part

After a first introductory part, we have chosen to separate the two main themes of our work which are Computer Aided Product Design in part B and decision making process for chemical product substitution in part C. Our CAPD tool and our decision making process are complementary, in the sense that they provides each other useful information. However, they can also be used independently: it is possible to use the CAPD tool in another context, and the decision making process can rely on another chemical product design method. Finally the epilogue sets the perspectives.

FIRST PART: INTRODUCTION

This part outlines the scientific context in which our work is performed, our objectives and the expected benefits of our work. Then the research frame of our work is described and all the themes that are addressed are presented. Finally the structure of the document is detailed and the different parts are described.

SECOND PART: COMPUTER AIDED PRODUCT DESIGN IN CHEMICAL FIELD

We plan to develop an innovative and efficient Computer Aided Product Design (CAPD) method and tool dedicated to chemical product and compliant with today's needs: advanced mixture handling, molecular fragment management… A state of the art of the existing method has allowed the definition of an innovative CAPD method using a meta-heuristic search method coupled with a multi-leveled approach and relying on a graph based molecular representation. This method has then been implemented in a software prototype called IBSS thanks to a model driven approach.

THIRD PART: DECISION MAKING PROCESS FOR CHEMICAL PRODUCT SUBSTITUTION

With regard to the complex nature of product substitution issues in an industrial context, we wish to formalize a generic approach dedicated to address efficiently the situations where a product substitution is needed. We base our approach on concepts from different fields, i.e. model driven engineering, enterprise modeling, decision making processes and requirement management. These concepts are detailed first and the approach we propose is described just after. Finally, an industry related case study is presented. It illustrates how our decision making process and our CAPD tool can be used to find a greener solvent for printing facilities.

FOURTH PART: EPILOGUE

In this final part, a general conclusion presents the main outcomes and contributions of our work and ends with a discussion on the limitations and perspectives.

The manuscript ends with the bibliography and with the appendixes.

SECOND PART

B. COMPUTER AIDED PRODUCT DESIGN IN THE CHEMICAL FIELD

2.	State of the art	……..	13
3.	Proposition of a Computer Aided Product Design	……..	51
4.	Development of the IBSS tool	……..	69

ABSTRACT

We plan to develop an innovative and efficient Computer Aided Product Design (CAPD) method and tool dedicated to chemical product and compliant with today's needs: advanced mixture handling, molecular fragment management… A state of the art of the existing method has allowed the definition of an innovative CAPD method using a meta-heuristic search method coupled with a multi-leveled approach and relying on a graph based molecular representation. This method has then been implemented in a software prototype called IBSS thanks to a model driven approach.

CHAPTER 2

State of the art

Within the REACH regulation context, finding new molecules that are both environment and health friendlier has become a major issue for the chemical related industry. To cope with the challenging complexity of the problem, “trial and error” usual methods must be replaced by more efficient ones, like Computer Aided Molecular Design (CAMD) that can handle simultaneously functional, economical, health, safety and life cycle constraints.

In this chapter, after having defined Computer Aided Molecular Design (CAMD), the main features of CAMD methods are outlined. The different molecular representation models, numerical resolution methods and mixture performance evaluation are presented and discussed. We finally outline the few existing methods of Computer Aided Mixture Design and describe their limitations.

2.1 FUNDAMENTAL PRINCIPLES OF COMPUTER AIDED MOLECULAR DESIGN

The concept of Computer Aided Design emerged in 1983 (Gani and Brignole, 1983), in relation with progresses made in group contribution methods for estimating property values. The term of CAMD appeared three years later in Brignole et al. (1986). Many articles have proposed their own approach in several domain of application, as for example in the solvent design (Gani and Brignole, 1983; Pretel et al., 1994; Sinha et al., 1999; Cismondi and Brignole, 2004), the substitution of refrigerant fluids (Constantinou et al., 1996; Churi and Achenie, 1997; Vaidyaraman and Maranas, 1999) and the polymer design (Maranas, 1996, 1997).

The main goal of the CAMD is, for a given set of constraints on properties, to find the molecules built with functional groups that match the property constraints. It is the opposite of the traditional methods of “trial and error”. In these last methods, the chemist synthesizes molecules that, according to his experience and competence, might match the property constraints. These molecules are then evaluated and eventually eliminated if they do not fit to the expected constraints. The trial and error approach is time consuming, represents a consequent workload and is hence expensive. Further, it method is not adapted to multi-objective problems: the chemist’s solution may match a primary property but will fail to satisfy secondary constraints. On the other hand, CAMD can handle many constraints, provided that property estimation models are available.

The two methodologies are represented on Figure 3.

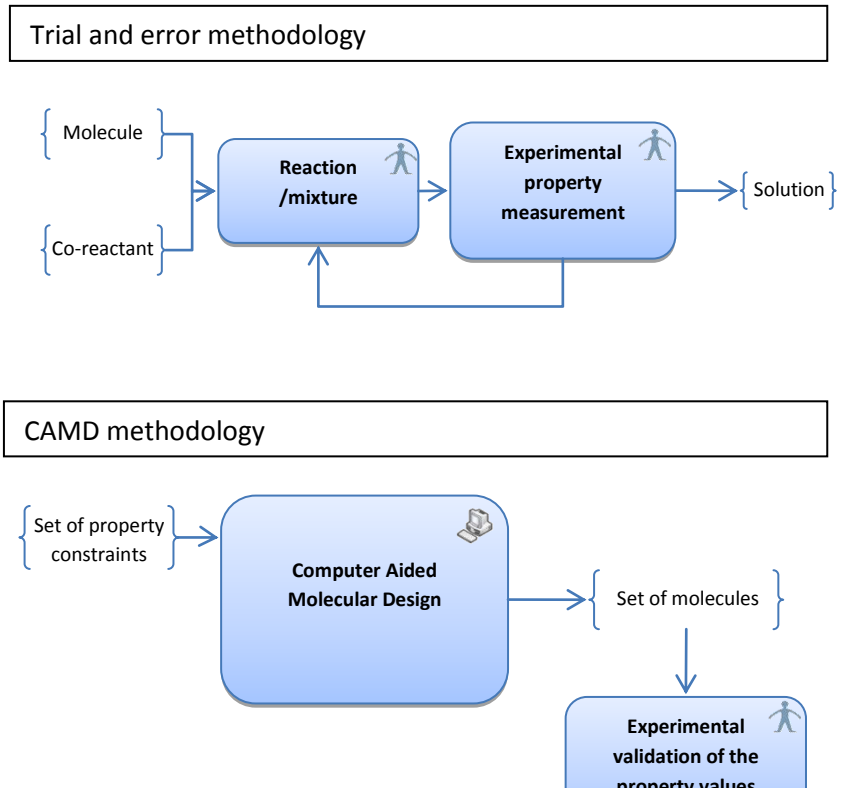


Figure 3: The trial and error and the CAMD methodology

Even though CAMD is a largely studied subject, there is no consensus on a unique formalized definition in the literature. According to (Korichi et al., 2008), “CAMD is a methodology of “inverse formulation” where target property values are first set and candidate molecules are sought among existing databases or constructed to satisfy the target values”.

We consider the following definition: CAMD is a methodology based on computer tools, to find molecules satisfying set of property targets that are defined in advance. We have identified two methods in the literature: the database approach and the group contribution based approach.

The database approach consists in consulting an existing database. On the contrary, the group contribution based approach is well studied and is at the origin of the term CAMD. In this method, the molecules are constructed with functional groups. They are evaluated thanks to group contribution methods and then discriminated in order to keep only the bests. This method implies, in addition to the definition of the property target values, the definition of the set of functional groups that are used for the molecular construction.

Moreover, it is based on four main principles:

- A molecular representation model
- A set of property calculation models
- A resolution method
- A set of performance criteria

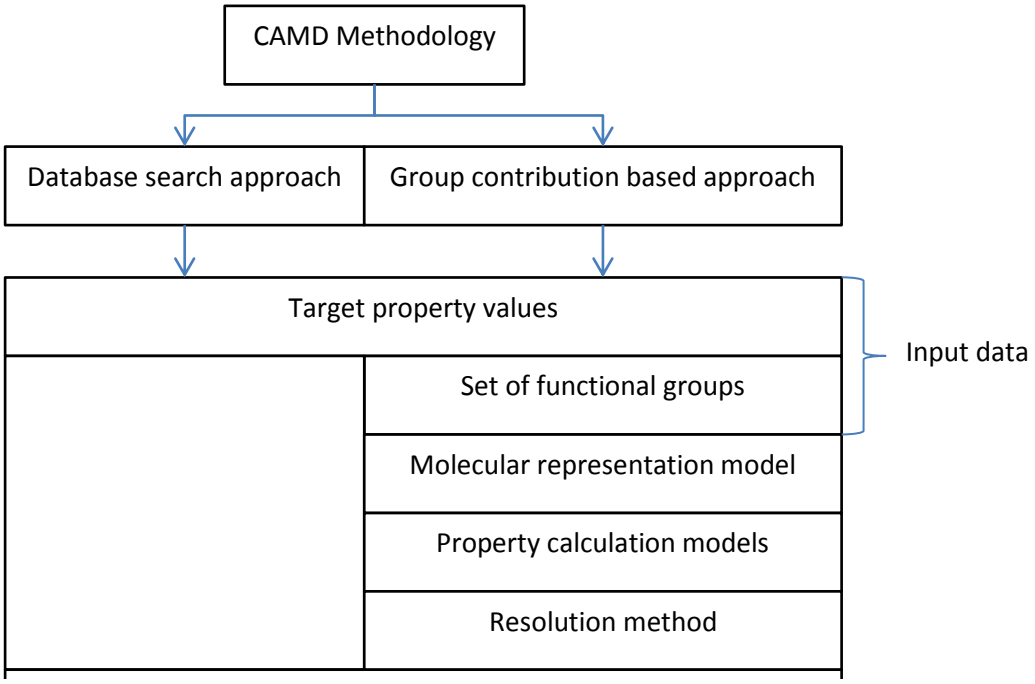


Figure 4: CAMD methods

In the following paragraphs, we survey the four principles enounced above.

2.2 MOLECULAR REPRESENTATION MODELS

The molecular representation is an important issue for CAMD because the property calculation methods that can be used depend on the molecular representation. With a complex representation, more calculation methods can be used but the combinatory complexity of the molecule structure becomes a computational challenge.

Very accurate representations of molecular structures exist, sometimes including chirality. SMILES “Simplified Molecular Input Line Specification” (Weininger, 1988; Weininger et al., 1989) and WLN “Wiswesser Line Notation” (Smith et al., 1968) are popular methods. However, those sophisticated

representations are difficult to handle and thus not fitted for CAMD, although more and more property calculation methods can use these notations as input.

Several molecular representations which have been used in CAMD are presented hereafter.

2.2.1 Strings

In their CAMD approach using genetic algorithms, Patkar and Venkatasubramanian (2002) use a string representation which consists in lists of groups. The first list is the backbone-chain. It is followed by lists that represent the side-chains in the order of the backbone-chain.

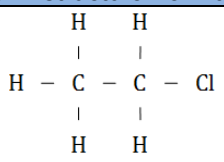
Chemical formula	Semi-structural formula	Computable representation model
CH3CH2Cl		((C C)((H H H)(H H Cl)))

Figure 5: Example of a string representation as used by Patkar and Venkatasubramanian (2002)

- (C C) represents the backbone-chain.
- (H H H) represents the three hydrogen atoms connected to the first carbon atom of the backbone-chain.
- (H H Cl) represents the three atoms connected to the second carbon atom of the backbone-chain.

This representation allows representing the molecular structure but does not permit to deal with cyclic structures.

2.2.2 Collections of groups

A collection of groups specifies the groups present in a molecule and their occurrence. There are many ways to represent it in a computable manner.

The enumerative method of Gani uses two vectors:

- SG which contains the identities of the functional groups present in the compound.
- NT which represent the number of occurrence of each group in the compound.

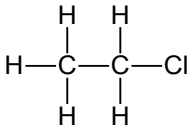
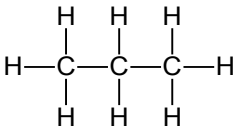
Chemical formula	Semi-structural formula	Computable representation model
CH3CH2Cl		$SG = [CH_3; CH_2; Cl]$ $NT = [\ 1\ ;\ \ 1\ ;\ 1\ ]$
C3H8		$SG = [CH_3; CH_2]$ $NT = [\ \ 2\ ;\ \ \ 1\ ]$

Figure 6: Examples of collections of groups as used in Gani's method

In the Simulated Annealing (SA) algorithm from Song and Song (2008), the two vectors of Gani are combined in a matrix with two rows.

Marcoulaki and Kokossis (2000a) proposed a similar representation with a group vector which contains the groups present in the molecule and a composition matrix which is a diagonal matrix containing the number of occurrences of the group in the molecules. The molecular vector is obtained by multiplying the vector with the matrix.

$$[2CH_3 \quad CH_2] = [CH_3 \quad CH_2] * \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \tag{1}$$

This type of representation is limited because the way the groups are connected to each other is not represented. Considering property estimations, only group contribution methods can be used and more specifically only those that are based on the same groups as the ones in the collection.

Such a representation is not unambiguous as a single collection represents all the isomers.

2.2.3 Binary representation

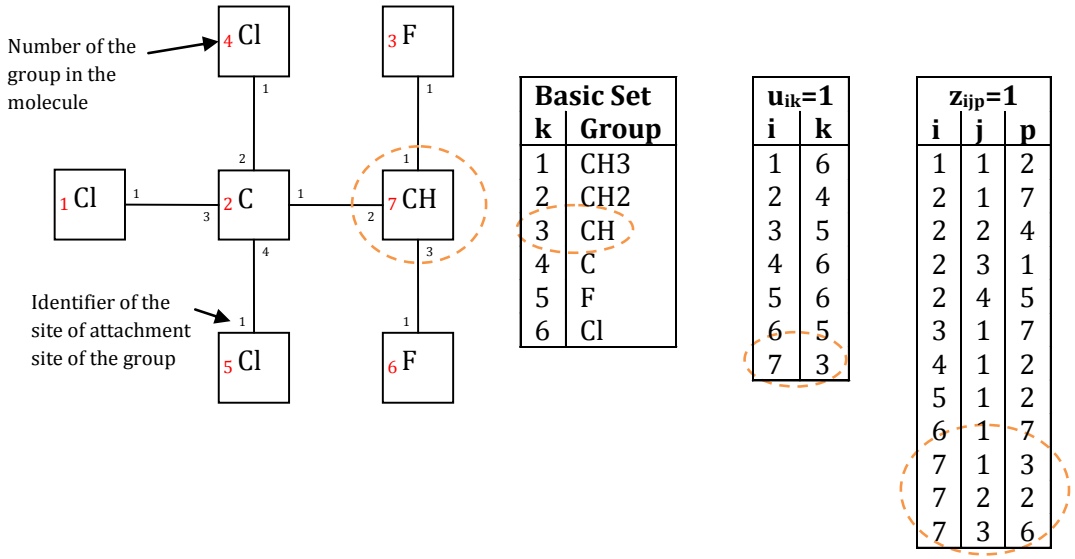
Churi and Achenie (1996) introduced a binary molecular representation. All the functional groups are stored in a list. All the attachments (called sites) of a group have an identification number.

The molecules are represented with a basis of groups and two connection indexes:

$$u_{ik} = \begin{cases} 1\ if\ the\ i^{th}\ group\ in\ the\ molecule\ is\ of\ the\ k^{th}\ kind \\ 0\ otherwise \end{cases} \tag{2}$$

$$z_{ijp} = \begin{cases} 1\ if\ the\ i^{th}\ group\ in\ the\ molecule\ has\ its\ j^{th}\ site\ attached\ to\ the\ p^{th}\ group \\ 0\ ohterwise \end{cases} \tag{3}$$

Here follows an illustrative example with 1,1,1-trichro-2-2-difluoroethane:



Basic Set	
k	Group
1	CH3
2	CH2
3	CH
4	C
5	F
6	Cl

u_{ik}=1	
i	k
1	6
2	4
3	5
4	6
5	6
6	5
7	3

z_{ijp}=1		
i	j	p
1	1	2
2	1	7
2	2	4
2	3	1
2	4	5
3	1	7
4	1	2
5	1	2
6	1	7
7	1	3
7	2	2
7	3	6

Figure 7: binary representation of 1,1,1-trichro-2-2-difluoroethane (from Churi and Achenie, 1996)

If we consider the group "CH", it corresponds to the 3rd group in the Basic Set list and it is the 7th group in the molecule. That's why u_{73} is equal to 1. This group has three attachment sites. The 1st is connected to the 3rd group in the molecule ($z_{713} = 1$), the 2nd to the 2nd group in the molecule ($z_{722} = 1$) and the 3rd to the 6th group in the molecule ($z_{736} = 1$).

This representation is well adapted for a mathematical optimization but is more difficult to apprehend for the user. The characterization of the sites and the variable z make it difficult to use this representation in random search methods.

2.2.4 "Classic" Graphs = G(X, V)

Classic graphs are represented with two sets:

- The vertex set $v = \{1, 2, \dots, N\}$
- The edge set $\xi = \{(i, j) | vertices\ i\ and\ j\ are\ connected\ by\ an\ edge\}$

Raman and Maranas (1998) introduced these concepts in CAMD and used the vertex adjacency matrix representation. The molecular representation consists of two variables:

- A vector containing the vertex i.e. the functional groups
- The adjacency matrix A where:

$$a_{ij} = \begin{cases} 1\ if\ the\ vertex\ i\ is\ connected\ to\ the\ vertex\ j \\ 0\ otherwise \end{cases} \quad (4)$$

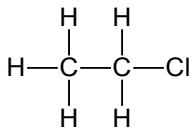
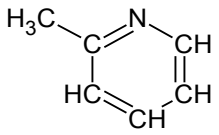
Chemical formula	Semi-structural formula	Computable representation model
CH3CH2Cl		$\begin{pmatrix} CH3 \\ CH2 \\ Cl \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
C6H7N		$\begin{pmatrix} N \\ CH \\ CH \\ CH \\ CH \\ C \\ CH3 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

Figure 8: Examples of "classic" Graph representations as used by Raman and Maranas (1998)

This representation is easy to understand and easy to compute. But the fact that there are no differences between the types of connections makes it difficult to interpret and it is sometimes impossible to differentiate two molecules as shown on Figure 9.

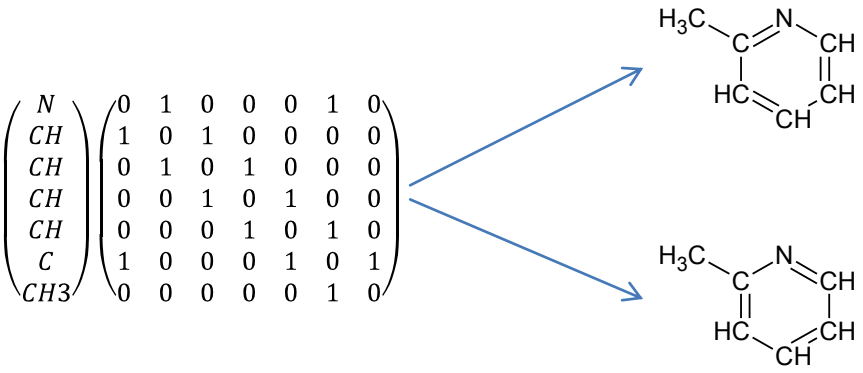


Figure 9: lack of precision of the graph representation model

(Lin et al., 2005) proposed a more complicated adjacency matrix representation. They replaced the 2 dimensions matrix A by the 3 dimensions matrix F where:

$$f_{ijk} = \begin{cases} 1 \; when \; the \; basic \; groups \; i \; and \; j \; are \; bonded \; with \; a \; k^{th} multiplicity \; bond \\ 0 \; otherwise \end{cases} \quad (5)$$

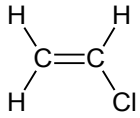
Chemical formula	Semi-structural formula	Computable representation model
CH2CHCl		$f_{ij1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ $\begin{pmatrix} CH2 \\ CH \\ Cl \end{pmatrix} f_{ij2} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $f_{ij3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

Figure 10: Example of the molecular representation used by Lin et al. (2005)

This representation is a solution to the limitations of the previous representation but introduced in the meantime too many degrees of freedom. For example, this representation allows two groups to be connected by more than one bond if these bonds are not of the same multiplicity. This imposes to take into account additional mathematical constraints on the matrix.

2.2.5 Structure-composition matrix

In their work, Ourique and Silva Telles (1998) proposed a matrix representation where the information about the vertices and the edges is combined in a single matrix A:

$$a_{ij} = \begin{cases} if\ i \neq j, & a_{ij} = \begin{cases} 1\ if\ the\ group\ i\ is\ connected\ with\ the\ group\ j \\ 0\ otherwise \end{cases} \\ if\ i = j, & a_{ij} = unambiguous\ identifier\ of\ the\ group\ i \end{cases} \quad (6)$$

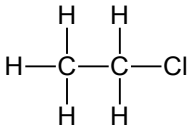
Chemical formula	Semi-structural formula	Computable representation model
CH3CH2Cl		$\begin{pmatrix} CH3 & 1 & 0 \\ 1 & CH2 & 1 \\ 0 & 1 & Cl \end{pmatrix}$

Figure 11: Example of the matrix representation of Ourique and Silva Telles (1998)

Combining the two pieces of information into a single variable makes it more readable and easier to compute. But in this representation, the lack of connection characterization remains a problem.

2.2.6 Signature

A Signature representation is used by Weis and Visco (2010). The molecule is represented with its atomic Signature at height 1 which consists of the list of all atoms of the molecule associated to their direct neighbors. The height determines the level of observation: at height 0 no neighbor is considered, at

height 1 the direct neighbors are considered, etc. The molecular Signature is the sum of all atomic signatures. An illustrative example is given on Figure 12.

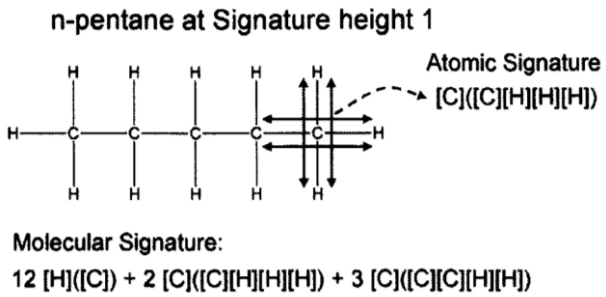


Figure 12: Signature translation example (from Weis and Visco, 2010)

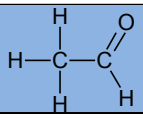
2.2.7 Others

Van Dyk and Nieuwoudt (2000) used a chromosome-like representation. The molecule is represented as a linear combination of genes where genes are a non-necessarily linear combination of structural groups. Very little information is given on this technique, and none on how the genes are constructed.

2.2.8 Conclusion

The molecular representation models used in CAMD are numerous and very different. The main ones are summed up in the table here after.

Table 1: Summary table of the commonly used molecular representation in CAMD

Type	reference	Example 	Comments
String representation	(Paktar and Venkatasubramanian , 2003)	((CC)((HHH)(HO))	Very precise but gets difficult to read when the molecule is complex. It is also not easy to compute.
Collection of groups	(Gani et al., 1991)	$SG = [CH3; CH; O]$ $NT = [1;\ 1;\ 1]$	Easy to understand and to compute but it gives no detail on how the groups are connected.
Collection of groups	(Marcoulaki and Kokossis, 2000)	$[CH3; CH; O] * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	ditto
Binary representation	(Churi and Achenie, 1996)	Basic set = {CH3-, -CH=, O=} $\{u_{ik}=1\}$={ (1,1);(2,2);(3,3)} $\{z_{ijp}=1\}$= {(1,1,2);(2,1,1); (2,2,3);(3,1,2)}	Precise but not intuitive and not easy to handle.
Graph representation	(Raman and Maranas, 1998)	$\begin{pmatrix} CH3 \\ CH \\ O \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	Easy to compute and to understand but suffers from the lack of precision on the type of the connections.
Graph representation	(Lin et al.,2005)	$f_{ij\,1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ $\begin{pmatrix} CH3 \\ CH \\ O \end{pmatrix} f_{ij\,2} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ $f_{ij\,3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	Precise and easy to compute, but has too many degrees of freedom.
Structure composition matrix	(Ourique and Telles, 1998)	$\begin{pmatrix} CH3 & 1 & 0 \\ 1 & CH & 1 \\ 0 & 1 & O \end{pmatrix}$	Easy to compute and to understand but suffers from the lack of precision on the type of the connections.
Signature	(Weis and Visco, 2009)	4[H]([C])+[C]([C][H][H][H]) +[C]([C][H][O])+[O][C]	Very precise but gets difficult to read when the molecule is complex. It is also not easy to compute.

In our CAMD method, we will use a structure composition matrix to which we will add an attribute for the type of the connections. We believe it is a good representation because it is easy to understand and to handle for a computer program while remaining very accurate.

2.3 NUMERICAL RESOLUTION METHODS

To resolve a combinatorial problem, many numerical methods exist. Three groups of methods can be identified: the exhaustive methods, exact methods and meta-heuristic search methods. Methods of the three groups have been used or adapted to CAMD. They are presented below.

2.3.1 Exhaustive methods (Generate and Test)

The CAMD exhaustive methods are also called “generated and test CAMD” and are based on the generation of a large number of molecules which are then tested to check if they are compliant with the constraints. In order to limit the combinatory explosion, the generation is based on the classification of the functional groups and on rules of connection between groups. The three main methods are presented.

2.3.1.1 Method of Gani and colleagues

In the method defined by (Gani et al., 1991), the UNIFAC groups are classified in class, according to their valency, and in categories, according to the limitation of their presence in the molecule following chemical feasibility rules, as shown on Table 2. This allows limiting the number of molecules that can be constructed by considering only chemically feasible molecules. The properties are classified into explicit properties (prediction methods available) and implicit properties (prediction methods not available, their value has to be found by experiments or in literature).

Table 2: Classification of the Groups by (Gani et al., 1991)

	Category				
Class	1	2	3	4	5
0	CH3OH CH3SH (CH2OH)2 NMP Diethyl Glycol 2-propanol CCL2F2	CH3NO2 CH3CN CH2CL2 CH3NH2 CCL3F C4H4S	H2O Furfural CHCL3 TCE Pyridine CHCL2F Morpholine	CH3NH2 HCOOH ACRY MFA 1-propanol CHCLF2	CF3 DMSO DMF TMS CS2 CCLF3
1	CH3	CH2CN CH2NO2 CH2NH2	CH3CO CONH2 CONHCH3 CON(CH3)2	OH CHO COOH CH2CL I, Br F, CL CH3COO CH3O C2H5O2 CH3S	CCL2F CH2SH CH3NH CHCL2 C4H3S SH C≡CH COO CCL3 SiH3 CH2NH2 CCL2F CHCLF
2	CH2	CHNO2	CH2CO CH2COO CH2O CONCH3CH2	CHNH2 CH2NH CHCL CONHCH2 C2H4O2 CH2S	CH=CH CH2=C CH4H2S CH3N C≡C SiH2
3	CH		CON(CH2)2	CHNH CH2N CCL CH–O CHS	CH=C CCL2 SiH SIH2O SiHO
4	C				C=C SiO Si
5	ACH		ACCH2 ACCH AC	ACCH3	ACOH ACNH2 ACCL ACNO2

*For class zero groups, the categories do not apply.

The method consists of four stages:

- **Preselection of the groups and the properties**

  Functional groups are selected (e.g. −CH3, −CH2−, −CH<, >C<). They will be used to construct the molecules.

  The properties and the target values (a minimum and a maximum value for each property) are chosen in order to correspond to the problem (e.g. boiling point, vapor pressure).

- **Generation of the feasible molecules**

  A set of rules is used in order to reduce the size of the combinatory problem. Among them, constraints on the chemical feasibility (valency must be equal to zero and the size of the molecule is limited). Other rules take into account the limitations of the group contribution methods. The molecules generated are represented as a collection of groups (e.g. {2CH3, CH2}).

- **Property prediction**

  The estimation of the explicit properties of the generated molecules is done thanks to group contribution methods like the model from Joback and Reid (1987) to compute Tc, Tb, etc. In this case the calculation equation can be formulated as $P = \sum_i N_i C_i$where P represents the property value, N_i represents the number of occurrences of the functional group i and C_i represents its contribution. The values are then compared with the targets. Only compounds that satisfy most of the constraints on properties are retained.

- **Final selection**

  The compounds are rated in terms of performance. The implicit property values are evaluated for the bests of them.

Constantinou et al. (1996) updated this method by introducing new group contribution methods, new groups and second order groups so that isomers could be differentiated.

Marrero and Gani (2001) completed with a third order contribution to account for the property value corrections due to group interconnections. These properties are melting point, boiling point, critical temperature, critical pressure, critical volume, Gibbs energy, enthalpy of formation, enthalpy of vaporization and enthalpy of fusion.

In order to be able to use more prediction models, Harper et al. (1999) used a multi-leveled molecular representation CAMD as illustrated on Figure 13. At each new level, the molecular representation is made more complex, enabling the use of new models. Then, all the compounds are evaluated and only the most fitted ones are retained as inputs of the next level. On the first level, UNIFAC group vector descriptions are considered. UNIFAC is a group contribution method for computing activity coefficient (Fredenslund et al., 1975). On the second level, the collections of groups resulting of level 1 are transformed into chemical structures by using a structure generation algorithm. On the third level, the selected chemical structures are written as a matrix describing the compound connectivity at the atomic level. On the fourth level, a 3D representation is used.

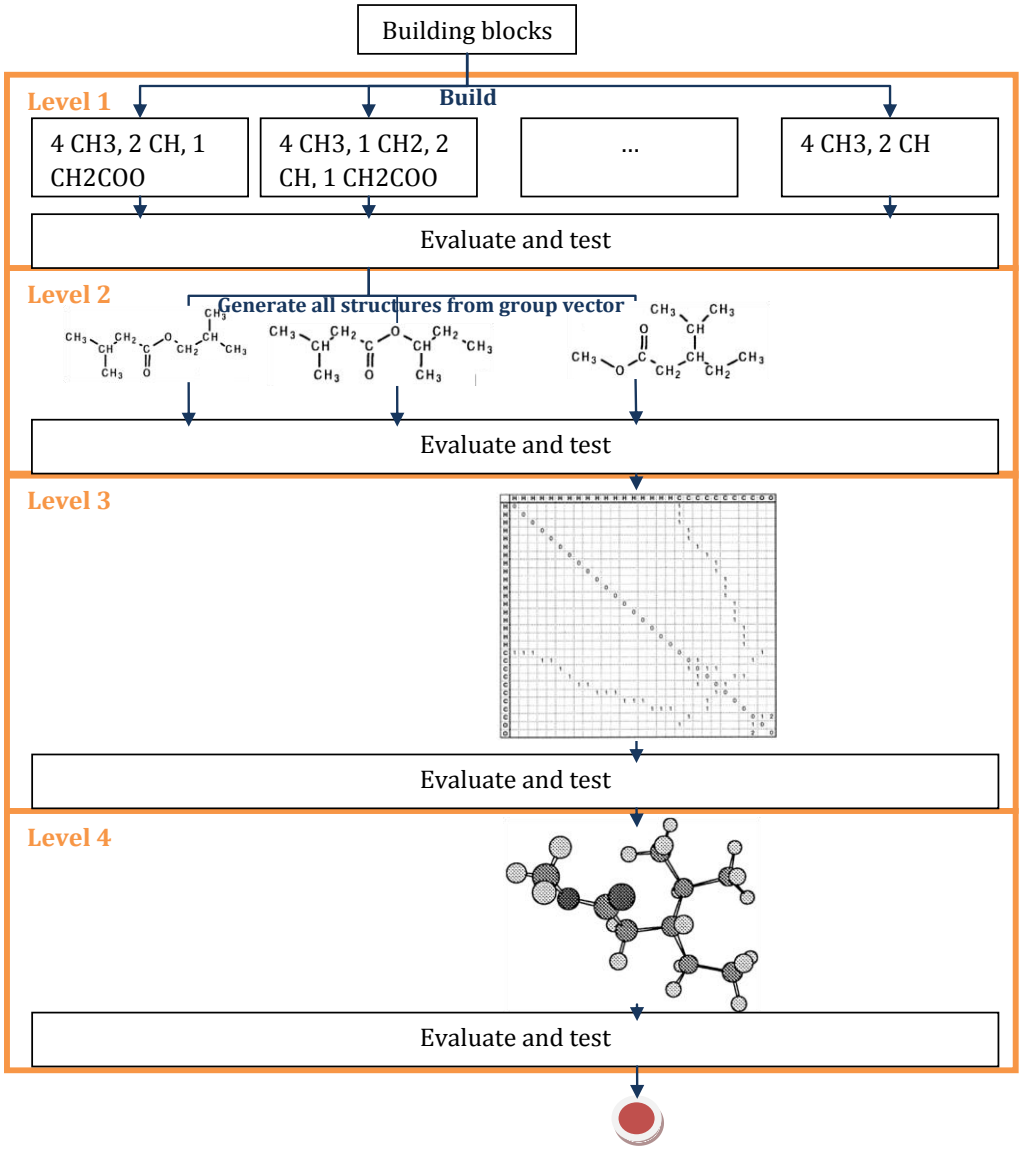


Figure 13: Harper et al. multi-leveled molecular representation algorithm

Each level detail is suited to different type of prediction models, from group contribution at level 1 to molecular simulation at level 4.

Another multi-leveled approach proposed by Korichi et al, (2008). It is based on a detailed molecular graph. Molecular graphs contain information precise enough to be used by a large variety of property estimation models. As seen in Figure 14 taken from Korichi et al. (2008), a molecular graph (top left) is well suited for the kind of property estimation methods listed on the right. It can even be expanded with explicit hydrogen into an all atom matrix which could be fed as a starting point to some molecular simulation packages.

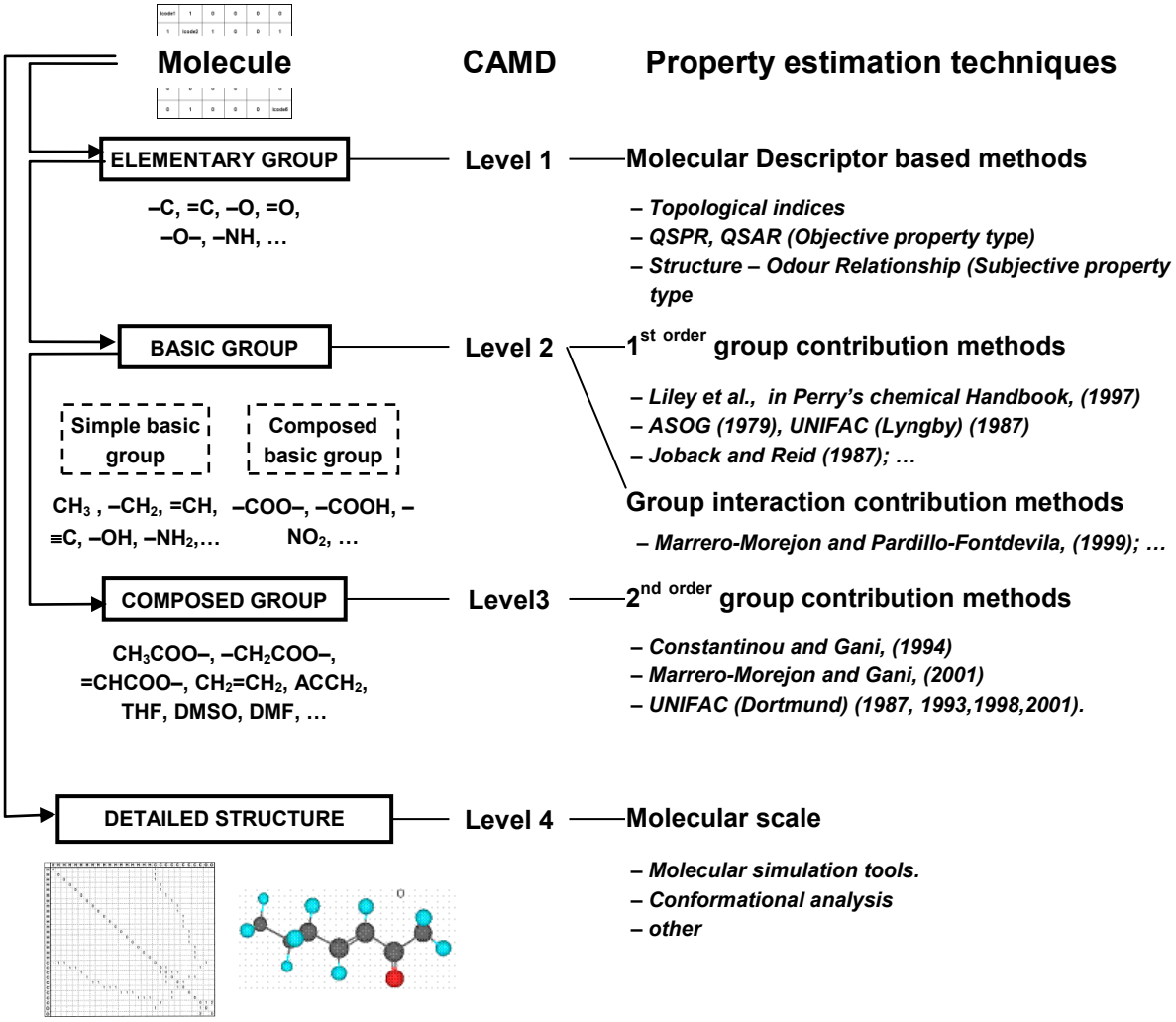


Figure 14: Molecular graph adaptation to property estimation methods (from Figure 3 in (Korichi et al., 2008))

2.3.1.2 Method of Brignole et al

The method of (Gani and Brignole, 1983), (Brignole et al., 1986) and (Pretel et al., 1994) is an evolution of the first method of CAMD. It has been upgraded in order to reduce the combinatory problem

and to explore cyclic and aromatic molecules in addition to linear and branched ones. The functional groups are issued of the UNIFAC groups set. The compounds are represented as a collection of functional groups.

The generation proceeds in two stages. Firstly, intermediate molecular structures are generated. Those structures consist of sets of groups of valency of 2 or more which forms the central part of the future compound. Secondly, after a pre-screening which takes into account the chemical feasibility and the models ability to evaluate their properties, terminals groups (groups of valency of 1) are added. The properties of those new structures are then evaluated and the best molecules are selected.

The major issue of evolution is the combination rules (i.e. the feasibility restrictions). The use of combination rules is a mean to prevent the formation of unstable compounds. At first, those rules were only considerations on attachments of non-hydrocarbon groups, like “all non-hydrocarbon groups can only combine with a carbon attachment”. Then these considerations evolved to become a real characterization of the UNIFAC groups according to their attachment. For example it can be “severely restricted”, “partially restricted” or “unrestricted”. This leads to more precise feasibility criteria based on the number of each attachment type in a molecular structure.

2.3.1.3 Method of Joback and Stephanopoulos

The method of Joback and Stephanopoulos (1995) is an iterative method that relies on the abstraction of the molecular representation in order to deal with the combinatory explosion. At each iteration, the level of abstraction is reduced and the molecular structures are tested and then screened out. The method consists of 5 steps:

1. **Transformation of the target properties**

   All properties are decomposed in order to deal only with group contribution methods.

2. **Generation of metagroups**

   The functional groups used are Joback and Reid’s. They are aggregated into large cluster of groups of the same valency which are called metagroups. For each metagroup, the interval of value of each group contribution method is determined (by considering the minimum and maximum contribution value of the groups of the metagroup).

3. **Generation of metamolecules**

A metamolecule is a collection of metagroups which is represented here as a vector of occurrences of each metagroup. Considering a minimum and a maximum number of groups, all the combinations (metamolecules) are generated.

4. **Test of the metamolecules**

   The structure of the metamolecules is tested against chemical feasibility. The octet rule is used on the first iteration; other constraints, like constraints on the cyclic and aromatic group or on bond types, are used when relevant for the metagroups. If a metamolecule does not respect the rules, it is removed. On the remaining metamolecules, the interval of property value is calculated from the aggregation of the intervals metagroups making the metamolecule. If the intersection of the calculated interval and the property target interval is empty then the metamolecule is eliminated.

5. **Reduction of the abstraction**

   One metagroup is then divided in two new metagroups separating the high contribution groups and the low contribution groups. The new intervals are calculated. The remaining metamolecules are expanded in order to integrate the new metagroups. All combinations are generated.

The two last steps repeat themselves until all metagroups contain only one group. Then the test is done one last time and the remaining molecules are the solution of the considered problem.

Exhaustive methods are limited because of the combinatorial explosion. Feasibility rules are integrated to reduce it but, as Marcoulaki and Kokossis (1998) state without benchmark, they might also bias the algorithms towards traditionally used materials.

2.3.2 Exact methods

An exact method is a resolution algorithm which will always find the best solution and will always behave predictably: for a given input, it will always carry out the search the same way and will give the same output. A specific exact method cannot solve every type of problems. Thus the first step is to model the problem, determine its type and finally choose the most adapted algorithm.

For CAMD, the problem to solve can be seen as an optimization problem subject to constraints set by the properties and the feasibility rules. This problem can be expressed with the two different manners which are presented in the following parts. In the first part, Mixed-Integer Non Linear Programming (MINLP) models are used and solved with different methods. In the second part, several Mixed-Integer Linear Programming (MILP) models and deterministic resolution methods are presented.

2.3.2.1 Mixed Integer Non Linear Programming based CAMD approach

Several works have been done in the use MINLP deterministic methods in CAMD like Odele and Macchietto (1993), Duvedi and Achenie (1996), Churi and Achenie (1996) and Vaidyanathan et al. (1998). Each one of these methods has been used on a well-defined application domain.

The first research work has been done by Odele and Macchietto (1993). The molecules are represented as a collection of functional groups by a vector containing the occurrences of the groups in the molecule. The structural feasibility constraints used are the octet rule and rules on a limitation of the number of groups. The problem is solved using an algorithm based on the augmented-penalty/outer-approximation (AP/OA) algorithm. This algorithm involves solving a finite sequence of nonlinear programming (NLP) subproblems and a mixed-integer linear programming (MILP) master problem. The solution of the relaxed problem which has continuous optimization variables (the numbers of groups in a molecule may not be integers) is used as an initial guess.

Duvedi and Achenie (1996) have used a similar approach applied to refrigerants.

A more complex molecular representation has been introduced by Churi and Achenie (1996). They used a set of binary variables in order to represent the connectivity between the groups. This induced different structural feasibility constraints which allow the use of more complex, and usually more accurate, group contribution techniques. The resolution method used is the augmented penalty-equation relaxation-outer approximation (AP/ER/OA) which is similar to the AP/OA.

In the works of Vaidyanathan (Vaidyanathan et al., 1998), polymer composite products are considered. They proceed in two stages. In the first stage, the optimum composite matrix elastic properties and other target properties of the wanted polymer, like the best fabric architecture and best fiber volume fraction, are determined. In the second stage, the matrix properties and the other matrix physical properties are used as target properties to the molecular design. They use the UNIFAC groups and vector of occurrences in the polymer repeating unit for the molecular design. The structural feasibility

of the polymer is provided by the structural feasibility of the monomer. The optimization problem is solved using the solver GINO (Liebman et al., 1986).

2.3.2.2 Mixed-Integer Linear Programming based CAMD approach

Maranas (1996) used a method based on mixed-integer linear programming. The objective function and the constraints (MINLP problem) are transformed into a MILP problem. The constraints on the chemical feasibility take into account the octet rule, considerations on the connections between the groups and the number of the different groups in the molecule. The molecules considered are polymers and are represented as a vector of occurrences of groups in the polymer repeat unit. The resolution method is GAMS/CPLEX (Brooke et al., 1988).

Raman and Maranas (1998) reused the fundamental concepts of this method and applied them to non-polymer molecules which were represented with their topological indices. Adapted structural feasibility rules were used.

The complexity of the problems to solve makes it difficult to expect anything more than a limited success of mathematical programming methods. Nevertheless, these methods can be efficiently applied to simplified and small size cases (Marcoulaki and Kokossis, 1998).

2.3.3 Meta-heuristic methods

Contrary to the exhaustive and exact methods, meta-heuristic search methods do not guarantee to give the best solution but propose to give relatively quickly a good one. They consist in iteratively improving the candidate solution. In CAMD, they are based on the probabilistic evolution of the solutions and thus can be qualified as stochastic or random search methods. Most of them are inspired by different natural and physical phenomena. The principal works of CAMD based on random search methods are now discussed.

2.3.3.1 Simulated annealing

The simulated annealing algorithm is based on the analogy between optimization problems and statistical physics. Modification operators are randomly and successively applied to transform an initial molecule into a molecule with a high performance. After each modification, the performance of the new considered molecule is evaluated. If the performance is higher than the previous molecule, then the new molecule is kept for further modifications. Otherwise an acceptance probability depending on the

difference of performance is used to determine if the new molecule is kept or is reconsidered. The acceptance probability usually follows the metropolis criteria (Metropolis et al., 1953):

$$B_{ij}(T) = \begin{cases} -\exp(\frac{\Delta F_{i,j}}{T}), & \Delta F_{i,j} > 0 \\ 1, & \Delta F_{i,j} \leq 0 \end{cases} \quad (7)$$

Where:

- $B_{ij}(T)$ is the acceptance probability for the move from state i to j under T.
- $\Delta F_{i,j}$ is the gain of performance between the state i and j.
- $T$is the "annealing temperature", a statistical cooling parameter (Aarts and van Laarhoven, 1985) that can be updated by a cooling schedule.

The main steps of a simulated algorithm are presented on Figure 15.

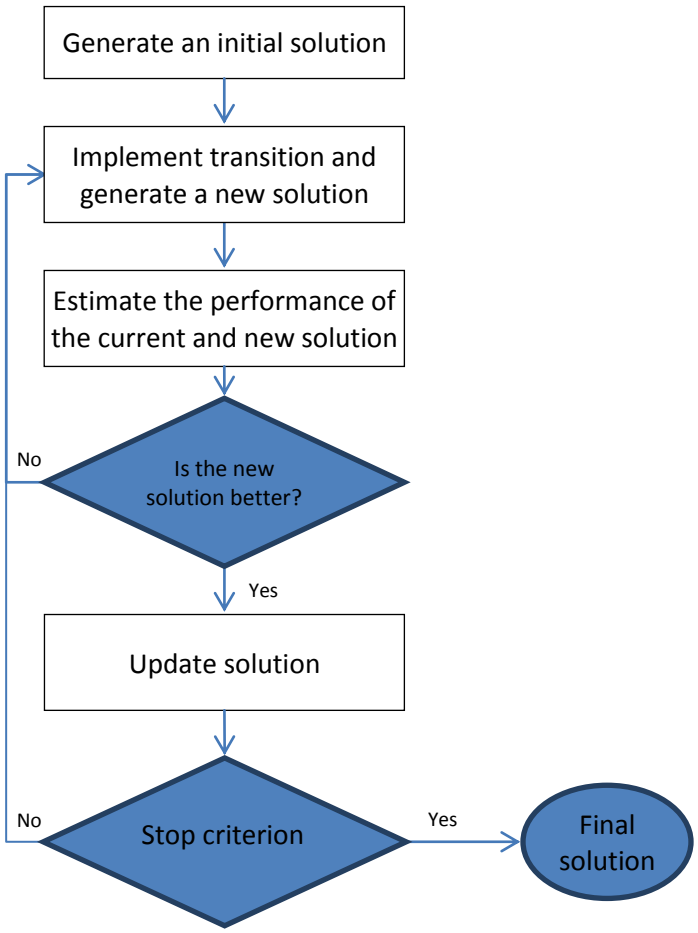


Figure 15: General steps of a Simulated Annealing algorithm

The real challenge is to find how to represent the molecules, how to define the modification operators and what the objective function is.

In Marcoulaki and Kokossis method (Marcoulaki and Kokossis, 1998) (Marcoulaki and Kokossis, 2000a), the molecular structures are represented by a collection of functional groups with a molecular and a composition vector. The modification operators are “substitution moves”, “expansion moves” and “contraction moves”. The objective function aggregates the synthesis objectives and the design constraints in terms of target property values and feasibility rules. An application of the method for the solvent design is presented in (Marcoulaki and Kokossis, 2000b).

Ourique and Silva Telles (1998) have defined the concept of a simulated annealing algorithm using molecular graphs. The molecular graphs are called structure-composition matrix and allow the representation of the connections between the groups. The modification operators proposed are “insertion”, “deletion”, “replacement”, “ring/aromatic insertion” and “deletion” and “ring fusion”. The objective function is an average function of the performance of each property. This performance is calculated with an exponential function.

Song and Song (2008) present an optimization CAMD approach based on a simulated algorithm. The groups used are the modified UNIFAC groups (Gmehling et al., 1993) and the molecules are represented as a collection of groups. The structural feasibility rules are based on the octet rules and on a constraint on the number of groups in the molecules. The modification operators are called transitions. They are the “insertion”, the “deletion” and the “replacement transition”. The multi-objective problem is converted into a single objective problem by keeping only one objective and converting the other into constraints.

2.3.3.2 Tabu search

The Tabu search algorithm explores the solution space by jumping from a solution to one of its neighbors and by using the Tabu list which contains the last solutions considered. If the new current solution is equal to a solution in the list, it is rejected and a different neighbor is considered as new current solution. This allows avoiding being trapped in local optima.

The main steps of a Tabu search are presented on Figure 16.

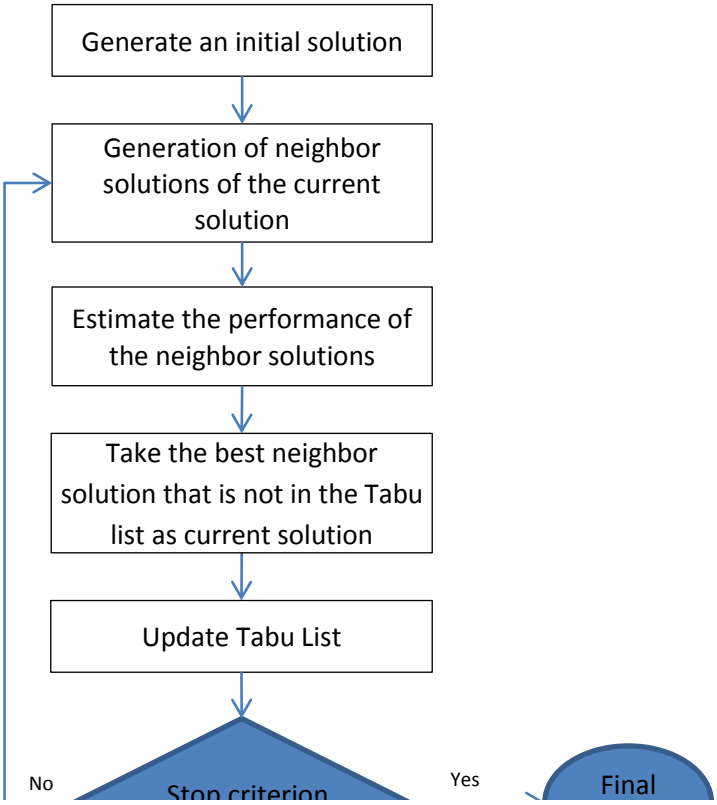


Figure 16: General steps of a Tabu Search

Lin et al. (2005) have developed a CAMD Tabu Search. Connectivity indices are used to represent the molecules. Structural constraints are considered to ensure that the molecules are fully connected and satisfy valency. The modification operators to get a neighbor are “replace”, “insert”, “delete”, “swap” and “move”. The objective function is the sum of the differences with the properties target values and is to be minimized.

2.3.3.3 Genetic algorithm

A genetic algorithm explores the solution space using the principle of natural selection and the laws expounded by Darwin. The fundamentals of this algorithm have been introduced by Holland (1975). A population of solution is successively modified using the best solution of the population n to create the population n+1. The population n+1 (children) is constituted of solutions of the population n (parent) which have been chosen using a performance biased method (often the Goldberg roulette wheel). They are modified afterwards thanks to the genetic operators “crossover” and “mutation”.

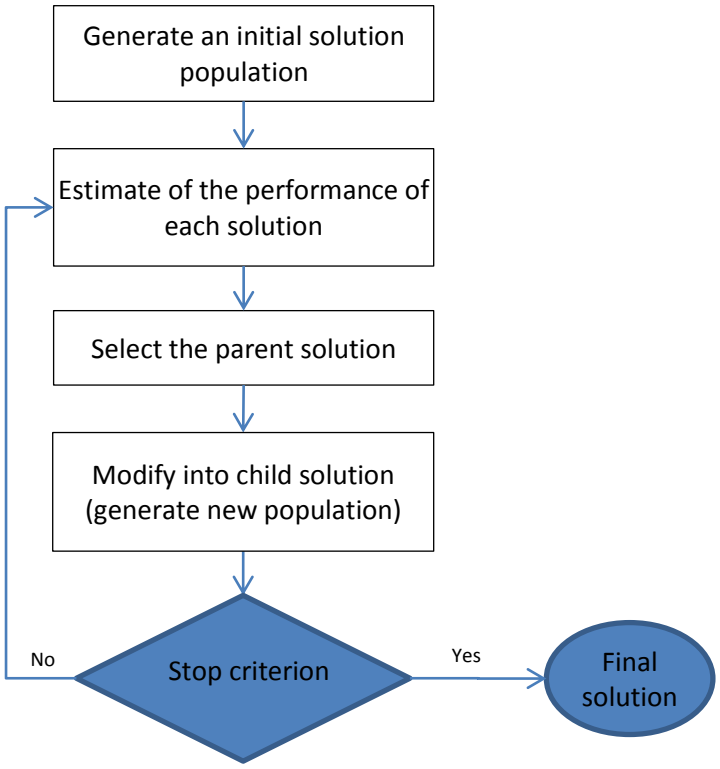


Figure 17: General steps of a Genetic Algorithm

The Genetic Algorithm has been first applied to CAMD by Venkatasubramanian and colleagues and it is summarized in Patkar and Venkatasubramanian (2002). The molecular representation is a string of symbols or functional groups which comprises a backbone-chain and side-chains. The modification operators are “single-point crossover”, “main-chain” and “side-chain mutation”, “insertion”, “deletion”, “blending” and “hop-mutation”. The objective function depends on the type of constraint considered: for target property value with some bounds, a Gaussian like function is used; for property constraints a sigmoid function is used.

Van Dyk and Nieuwoudt (2000) proposed a genetic algorithm based on a chromosome-like representation of the molecules using the UNIFAC groups. The algorithm has an elitism policy and four modification operators (“point mutation”, “crossover”, “insertion” and “deletion”). The objective function contained sigmoid fitness function.

2.3.4 Conclusion

Many different methods have been developed. They are summarized here after.

Table 3: Summary table of the search methods used in CAMD

Methods	References	Advantages	Drawbacks
Exhaustive methods	**(Gani et al., 1991)** (Constantinou et al. 1996) (Harper et al. 1999) **(Gani and Brignole, 1983)** (Brignole et al., 1986) (Pretel et al., 1994) **(Joback and Stephano-poulos, 1995)**	Easy to use on small problems Always convergent	Impossible to use on "real" problems".
Exact methods	(Odele and Macchietto, 1993) (Duvedi and Achenie, 1996) (Churi and Achenie, 1996) (Vaidyanathan and El-(Halwagi, 1998) (Maranas, 1996) (Raman and Maranas, 1998)	The formulation is standard. Proof of convergence. Well-tried methods. Available solvers and library.	Heavy formulation Long computation time
Meta-heuristic methods (Random searches)	(Marcoulaki and Kokossis, 1998) (Marcoulaki and Kokossis, 2000-a) (Ourique and Telles, 1998) (Song and Song 2008 (Lin et al., 2005) (Patkar and Venkatasubramanian, 2003) (Van Dyk and Nieuwoudt 2000)	Easy to use and implement. Efficient on specific problems	No proof of convergence Difficulty to tune the algorithm.

As we wish to handle complex molecular structures, we have chosen to implement a random search in order to have an acceptable computation time. We have, furthermore, chosen to start with the implementation of a genetic algorithm as some work had already been done on the subject in the team (Korichi et al., 2008).

2.4 PROPERTIES AND MODELS

CAMD is based on property estimation models. The quality of the results of the method depends largely on the accuracy of the models used. There exist various classes of property estimation models. Another distinction must be made for pure compound models and for mixture models.

2.4.1 Calculable and subjective properties

As recalled by Korichi (2010), we may distinguish properties with a numerical value and so-called subjective properties. Those later refer to properties described according to a scale related to the

observer perception: light is dim or bright, aroma is strong of faint. Once a numerical scaling of the perception is done, the subjective properties can be assimilated to calculable properties.

2.4.2 Pure compound property estimation models

Regarding pure compound models we can identify:

- Similarity methods: e.g. the octanol water partition coefficient Kow(aniline) = Kow(benzene) + f° (NH_2)
- Molecular descriptor based methods, like QSAR (quantitative structure activity relationship) or topological indexes methods. QSPR methods (quantitative structure property relationship) is another class dedicated to predict physico-chemical properties.
    - There exist thousands of descriptors, many of which have little physical or chemical significance. However they can be classified as 0D-descriptors (i.e. constitutional descriptors, count descriptors), 1D-descriptors (i.e. list of structural fragments, fingerprints), 2D-descriptors (i.e. graph invariants), 3D-descriptors (size, steric, surface and volume descriptors). Other descriptors like atomic signatures rely upon features of the molecular graphs themselves.
    - QSAR methods are usually built following a sequential strategy: (i) select a database of compounds and their properties; (ii) run a molecular descriptor software to compute the descriptor values; (iii) use statistical methods to identify the most significant descriptors; (iv) find the correlation that relates the property values to the descriptor occurrence in each molecule.
    - Many pitfalls must be avoided: like not representative enough database and statistical consistency. For that later, several methods are used: cross-validation test and training datasets, data randomization, etc...
    - The predictive value of these methods has been demonstrated for very complex problems, like HIV activity of new proteins, and explains its popularity among scientists. For less complex problems, like boiling point, its predictive capacity is usually below that of other methods, like group contribution methods.
- Group contribution methods relate the property value to the occurrence of several chemical sub-structures in the molecule. Each group is assigned a contribution to the property value, after usually running an optimization on some large set of data for known molecules.

- Simple groups are made from atoms with hydrogens, like -CH_3, -OH, =O, -NH_2, -COOH, etc. and are found in all CG methods. More complex groups are found in second and third order methods, considering central atoms and their neighbors or position of the –OH group on the carbon backbone for example. They usually help improve the predictive capacity as they can enable to distinguish position isomers like normal and secondary alcohols.
- A large database is needed and statistical consistency of the methods must be taken care when developing a group contribution based method.
- As being easy to understand due to their chemical description nature, these methods are very popular to estimate physico-chemical properties like phase transition temperature, viscosity, heat capacity, etc... Considering one of the best available group contribution method based on third order contribution with more than 200 first order groups, (Hukkerikar et al., 2012), it shows a predictive capacity which ranks from very good for boiling temperature (absolute average error of 6.17K for Marrero-Gani's Method (Marrero and Gani, 2001)) to poor for the melting temperature (absolute average error 15.99K). The calculation equation up to a third order contribution is:

$$P = \sum_i N_i C_i + W \sum_j M_j D_j + Z \sum_k O_k E_k \quad (8)$$

Where:

- P is the property value
- N_i is the occurrence of the i^{th} 1^{st} order functional group
- C_i is the contribution of the $i^{th}1^{st}$ order functional group
- M_j is the occurrence of the j^{th} 2^{st} order functional group
- D_j is the contribution of the $j^{th}2^{st}$ order functional group
- O_k is the occurrence of the $k^{th}3^{st}$ order functional group
- E_k is the contribution of the $k^{th}3^{st}$ order functional group
- W and Z are coefficients

Other group contribution methods exist like those based on group interaction contribution that accounts for interaction between groups or those based on atomic signatures (Weis and Visco, 2010).

2.4.3 Mixture property estimation methods

Mixture properties depend upon pure compound properties. They can show:

- A linear dependency (additive). e.g. molecular mass

$$P_{mixture} = \sum_{i=1}^{nc} z_i P_i \quad (9)$$

  Where z is either a molar fraction or a volumic fraction

- A nonlinear dependency, due to interactions among the mixture compounds:

$$P_{mixture} = P_{ideal} + P_{excess} \quad (10)$$

  Where P_{excess} is either a positive or a negative contribution. P_{ideal} is for example a linear dependency upon the pure compound properties, but can also be a more complex function. Typical cases of nonlinear property are mixture viscosity or mixture surface tension.

Some important nonlinear mixture properties can be inferred from process models. Those are indirect methods that require solving of some process model. Three examples:

- For the mixture boiling temperature, solving a vapor – liquid Flash calculation, with constant pressure and vaporization ratio equal to unity (flash P, ω=0), gives as a result the boiling temperature of the compound or mixture (Smith et al., 2000). Minimum or maximum boiling azeotropes are typical illustrations of the nonlinear dependency of boiling temperature.
- The calculation of mixture flash point that requires solving together a vapor – liquid equilibrium and an equation of saturation in the vapor phase (Liaw et al., 2011). Minimum or maximum flash points are typical illustrations of the nonlinear dependency of flash point.
- The solubility of a compound in the mixture requires solving a solid-liquid equilibrium. Minimum or maximum solubility behaviors have been occurring.

In order to be used for a great variety of situations, a CAMD tool should handle a large panel of property estimation models. We have seen that those models can be of very different types. In addition, the property estimation models are in a constant evolution. The design of a CAMD tool must take account the necessity to follow this evolution so that the tool remains up-to-date.

2.5 MIXTURE PERFORMANCE EVALUATION

The use of performance criteria is a mean to compare and rate the different solutions. It involves some performance functions which calculate the performance for a specific property and an objective function which aggregates the performances of all the properties.

2.5.1 Performance Function

Three types of objectives can be identified: (1) the target value objective for which the property value must be the closest to a specific value, (2) the property constraint for which the property value must be beyond a specific value (maximization or minimization) and (3) the range constraint for which the property value must be comprised between two specific values. The different performance functions found in the literature are presented in the following paragraphs.

2.5.1.1 Scaled deviations

The scaled deviation is used for calculating performance when target values are considered. Raman and Maranas (1998) and Lin et al. (2005) use it to calculate the performance of a property for a target value objective. It is formulated as:

$$Perf(M) = \frac{\left|Prop(M) - Prop_{target}\right|}{Prop_{target}} \tag{11}$$

Where:

- M represents the molecules considered
- $Prop_{target}$ is the target value for the property

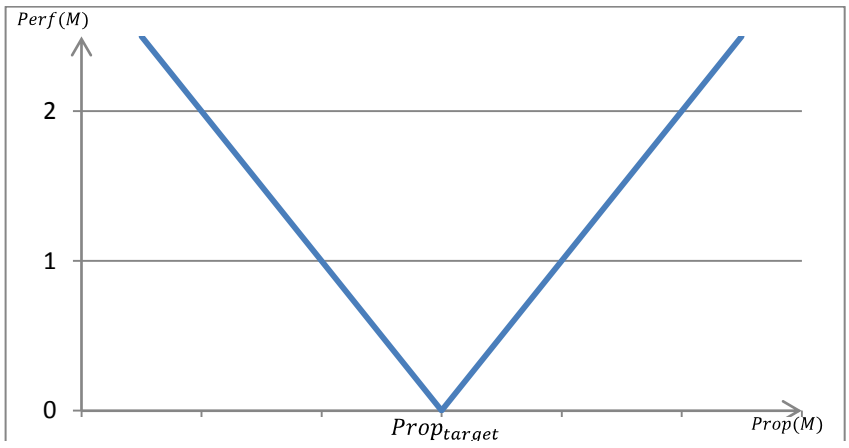


Figure 18: Performance calculated with a scaled deviation

The performance 0 corresponds here to a perfect match.

This performance calculation consists in assessing the distance between the property value of the molecule and the target, divided by the target value. For an optimization problem, the goal would be to minimize this performance. If only one property is to be optimized, then the division by the target value is useless. If several properties are to be optimized, then the divisions by the target values will not allow giving the same weight to each property in the optimization problem. Indeed, if one property has a target value ten times greater than a second property, then for a given molecule, the first property can be ten times further from its target than the second property and both would have the same performance.

2.5.1.2 Gaussian functions

Patkar and Venkatasubramanian (2002) use a Gaussian-like function to calculate the performance for a range constraint. It is formulated as:

$$\mathrm{Perf(M)} = \exp\left[-\alpha \frac{(\mathrm{Prop(M)} - \overline{\mathrm{Prop}})^2}{(\mathrm{Prop_{max}} - \mathrm{Prop_{min}})^2}\right] \quad (12)$$

Where:

- M represents the molecule considered
- α is a coefficient that determines the “width” of the function
- $Prop_{max}$ represents the maximum acceptable property value
- $Prop_{min}$ represents the minimum acceptable property value
- $\overline{Prop}$ is the average of $Prop_{max}$ and $Prop_{min}$ and represents the target value

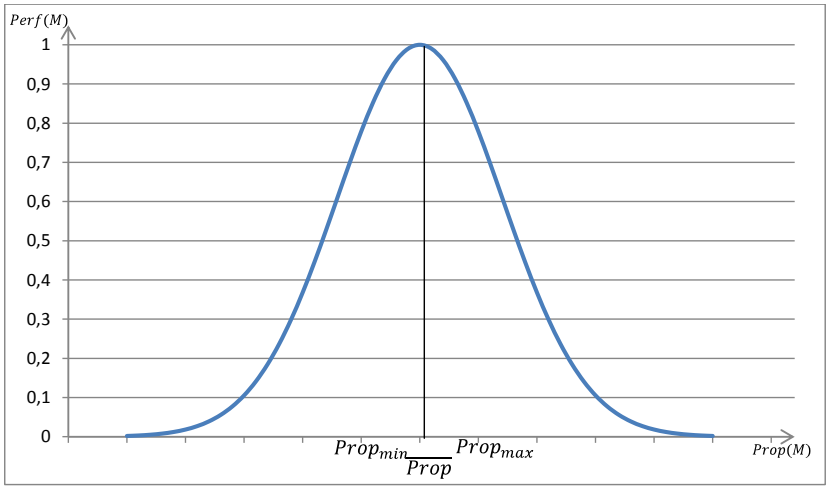


Figure 19: Performance calculated with a Gaussian function

This function only gives a number between 0 (excluded) and 1.

For an optimization problem, this performance is to be maximized. However, this results in selecting the molecules for which the property value is the closest to the center of the target range. This is not exactly the purpose for a range constraint performance calculation. Nevertheless, it can be adapted to the case of target value objectives.

2.5.1.3 Sigmoid functions

With sigmoid functions another type of target is considered: property constraints. The property value must be greater (or smaller) than a specific value. (Patkar and Venkatasubramanian, 2002) and (van Dyk and Nieuwoudt, 2000) use sigmoid functions in their genetic algorithm. Sigmoid functions used for a maximization can be formulated as:

$$\mathrm{Perf(M)} = \frac{1}{1 + \exp\left[-\beta \frac{\mathrm{Prop(M)} - \mathrm{Prop_r}}{\mathrm{Prop_r}}\right]} \tag{13}$$

Where:

- M represents the molecule considered
- β is a coefficient that determines the gradient of the function
- $Prop_r$ is the required property value, it corresponds to a 0.5 performance.

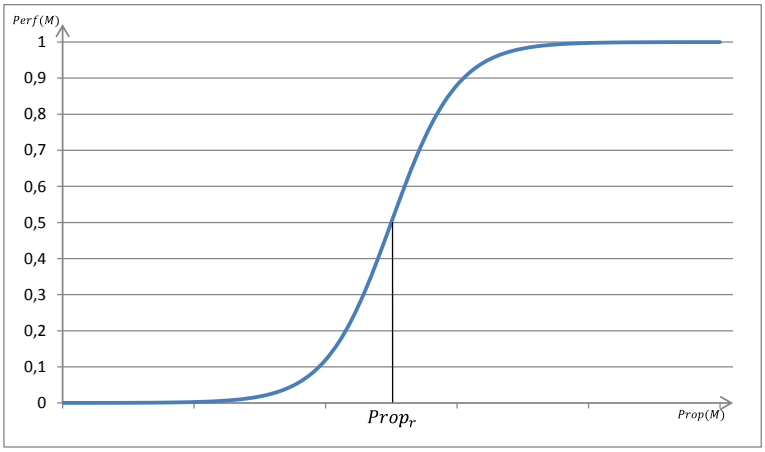


Figure 20: Performance calculated for a maximization with a sigmoid function

For a minimization, only the denominator needs to be modified. It is formulated as:

$$\mathrm{Perf(M)} = \frac{1}{1 + \exp\left[-\beta \frac{\mathrm{Prop_r - Prop(M)}}{\mathrm{Prop_r}}\right]} \tag{14}$$

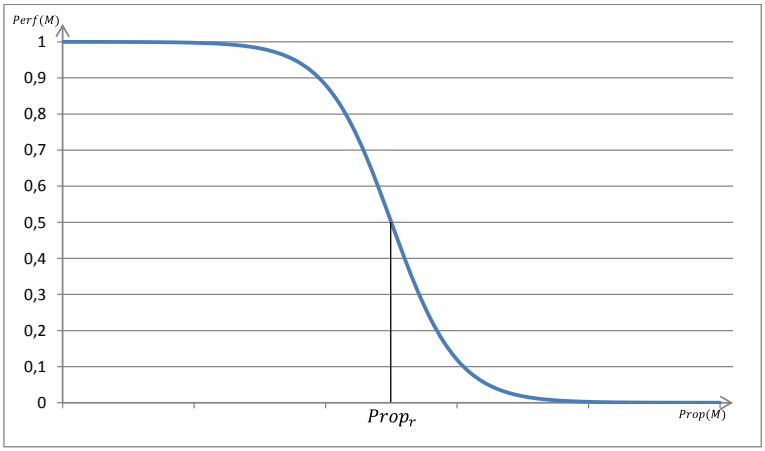


Figure 21: Performance calculated for a minimization with a sigmoid function

This function gives a number between 0 and 1 (both excluded). A result greater than 0.5 corresponds to a perfect match.

For an optimization problem, this performance is to be maximized. However, this results in selecting the molecules for which the property value is the furthest from the bound ($Prop_r$). This is not exactly the purpose for a property constraint performance calculation, since the solutions that are far from the bound will be better rated than those which are close, even though they are equivalent considering that they both meet the objective.

2.5.1.4 Desirability functions

Desirability functions are often used in random search method, but we haven't any references in the open literature that uses desirability function in CAMD. These functions can apprehend the three types of targets.

For a property constraint, the function is formulated as:

$$Perf(M) = \begin{cases} 0 & , \quad Prop(M) \leq Prop_{min} \\ \left[\dfrac{Prop(M) - Prop_{min}}{Prop_{max} - Prop_{min}}\right]^{r_i} & , \quad Prop_{min} < Prop(M) < Prop_{max} \\ 1 & , \quad Prop(M) \geq Prop_{max} \end{cases} \tag{15}$$

Where:

- M represents the molecule considered
- $Prop_{max}$ is the bound beyond which the function will return 1

- $Prop_{min}$ is the bound beyond which the function will return 0
- r_i is a coefficient modifying the function appearance

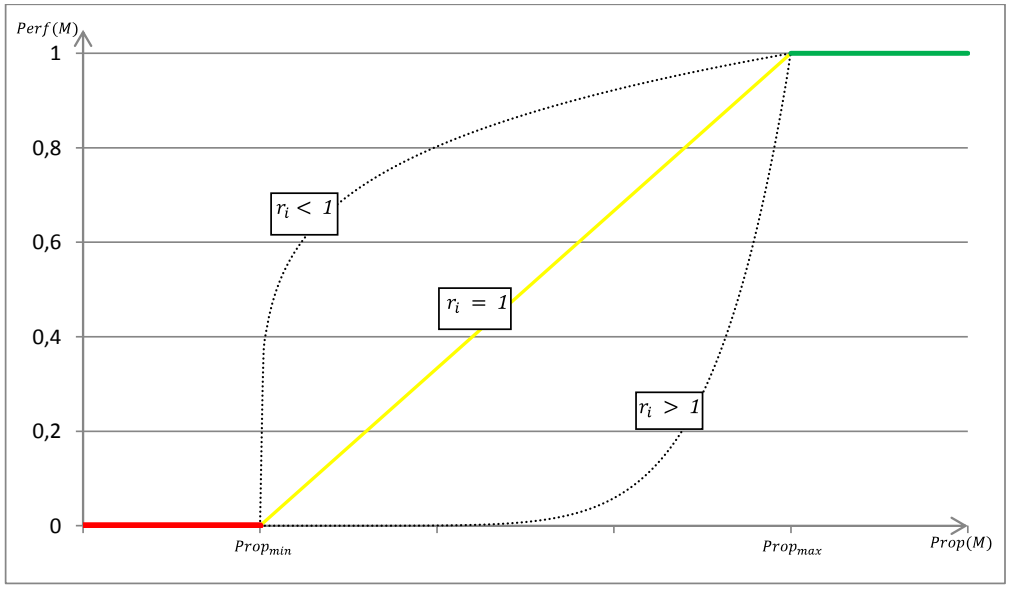


Figure 22: Performance calculated for a maximization with a desirability function

For the value target and range constraint, one simply has to combine two desirability functions, a maximization and a minimization.

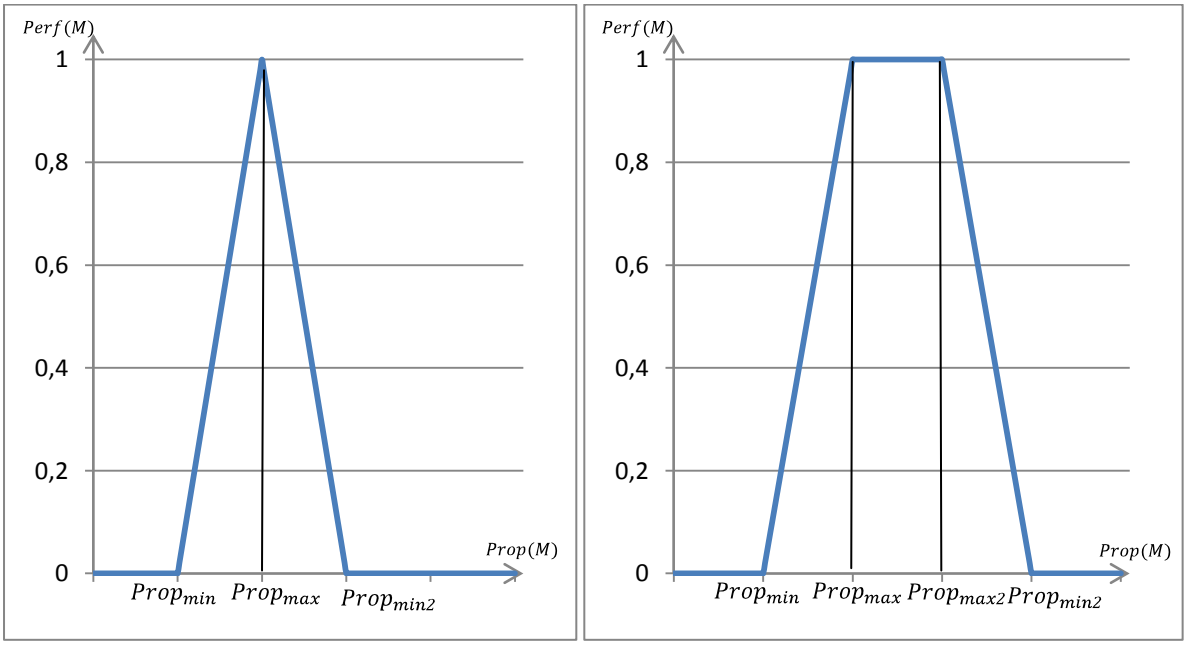


Figure 23: Performance calculated for a target value objective and a property constraint with desirability functions

For an optimization problem, this performance is to be maximized. This type of calculation does not present the problems of the Gaussian and the sigmoid function. Indeed, if two different molecules are a perfect match, they will be both rated with the maximal performance value no matter as far they are from the bound. The algorithm will then treat them as equivalent solutions.

2.5.2 Objective function

The problems of CAMD are often multi-objective problems. In order to transform the problem into a mono-objective one, the easiest way is to aggregate all the objectives into a single objective function.

In most of the works, the objective function is the average of performance functions. It usually has also weightings in order to be able to give more importance to certain objectives. This kind of objective function can be written as:

$$F(M) = \sum_{i=1}^{N} w_i * Perf_i(M) \quad (16)$$

- Mrepresents the molecule considered
- $N$is the number of objectives on properties
- w_i is the weighting of the ith objective.
- $Perf_i$ is the performance of the molecules for ith objective.

(Vaidyanathan and El-Halwagi, 1996), (Vaidyanathan et al., 1998) and (Patkar and Venkatasubramanian, 2002) use this representation. So does (Ourique and Silva Telles, 1998) but without the weightings.

Patkar and Venkatasubramanian (2002) use the product of all the performances calculated with a Gaussian function. It can be generalized as:

$$F(M) = \prod_{i=1}^{N} Perf_i(M) \quad (17)$$

This function penalizes more severely bad performances. Indeed, if one $Perf_i(M)$ is null then $F(M)$ is also null regardless of the potential good performances for the other objectives.

Another form of objective function is presented in (Raman and Maranas, 1998). It consists in:

- The maximal performance if the performance needs to be minimized (scaled deviation).

$$F(M) = \max_i Perf_i(M) \quad (18)$$

This objective function is then to be minimized.

- The minimal performance if the performance needs to be maximized (Gaussian function, sigmoid function, desirability function).

$$F(M) = \min_i Perf_i(M) \quad (19)$$

This objective function is then to be maximized.

2.5.3 Conclusion

As we wish to use the three types of objectives (target value, property constraint, range constraint), we have chosen to use the principles of combining different functions as done with the desirability function. The functions used can be varied: straight lines, Gaussian functions…

2.6 COMPUTER AIDED MIXTURE DESIGN

When the number of constraints increases, finding a pure compound that matches all the constraints becomes impossible. It is then necessary to consider mixtures (Churi and Achenie, 1997), (Duvedi and Achenie, 1997) and (Vaidyanathan and El-Halwagi, 1996).

Unlike Computer Aided Molecular Design, Computer Aided Mixture Design has not been widely investigated. The different methods of Computer Aided Mixture Design are presented hereunder.

2.6.1 Mixture design methods using existing molecules

(Klein et al., 1992), (Duvedi and Achenie, 1997), (Churi and Achenie, 1997) and (Sinha and Achenie, 2002) have adapted the CAMD mathematical programming approach presented in 2.3.2 to mixtures. The set of functional groups is replaced by a set of chemical compound and a vector of composition is added. The problem formalized as a MINLP problem and consists in finding the value of the vector of composition that corresponds at the best to the objectives. (Klein et al., 1992) solved the problem with a successive regression and linear programming (SRLP) algorithm. (Duvedi and Achenie, 1997) and (Churi and Achenie, 1997) uses an augmented penalty outer approximation (AP/OA) algorithm. Sinha and Achenie (2002) have implemented their own global optimization algorithm (LIBRA).

Conte and Gani (2011) proposed a chemicals-based formulation design software enabling virtual experimentations: PPD-lab Product-Process Design laboratory. Their method is based on a database approach. The user sequentially chooses:

- The active ingredient of the product in a database
- The solvents mixture in a list obtained by the MIXD algorithm (Conte, 2010). This algorithm generates all the combination of molecules of a specified database.
- The additives in a suggest list.

The MIXD algorithm is a multi-leveled enumerative database approach. On the first level, all the combinations of molecule of a specified database are generated. The linear constraints are evaluated and some combinations are rejected. For the remaining combinations, the optimal composition is calculated. It must respect the constraints and minimizes the cost. On the second level, the non-linear constraints are evaluated and mixtures that do not match these constraints are rejected. On the third level, the phase stability constraints are evaluated and mixtures that do not match these constraints are rejected. The remaining mixtures are presented to the user.

These methods propose to find the best mixture of already known compounds. There is no new compound created but it can be if the solvent molecules are generated using a CAMD tool.

2.6.2 Global approach

Vaidyanathan and El-Halwagi (1996) proposed a computer aided polymer blend design. In their method they optimize at the same time the molecular structure of the polymers and the composition of the mixture.

The molecular representation is based on general reaction schemes:

$$S - R - T + U - R' - V \rightarrow -[-X - R - Y - R' - Z] -_n \quad (20)$$

Where:

- S, T, U, V are molecular fragments determined a priori
- X, Y, Z are molecular fragments resulting from S, T, U, V
- R, R' are the molecular fragments to determine. They are represented as collection of groups

The feasibility constraints of the polymer are based on the feasibility constraints of the monomers.

In this method, only binary polymer blends are considered. This implies the addition of the parameter c1 which is the volume fraction of the first polymer in the blend (NB: c2=1-c1).

The problem is formalized as an MINLP problem and is solved thanks to the solver GINO.

2.6.3 Concluding remark

Most of the methods of mixture design propose to optimize the composition of preexisting molecules. We have found only one method in the open literature that proposes an approach similar to the one we wish to implement, which is to optimize at the same time molecular structures and optimization.

2.7 CONCLUSION

In this chapter, the different approaches of computer aided molecular and mixture design have been presented. They all consist in finding molecules that satisfy a list of property related constraints set initially. It emerges that the differences between the methods are numerous: e.g. on molecular representation models, on resolution methods and on performance criteria.

Molecular representation models are numerous: linear, string representation, collection of groups, binary representation, graph representation, structure composition matrix and signature. The most used in CAMD usually rely upon group vectors which are not unambiguously representing single molecules. They are also mostly suited for group contribution property estimation methods based on the same groups than the representation model. This drawback can be overcome by a multilevel frame used to generate all the molecules from the group vectors, then giving access to other types of property estimation methods.

Resolution methods used in CAMD can be classified as exhaustive, exact and meta-heuristic. The exhaustive ones are always convergent but are not suited for real problems where a combinatorial explosion may occur, unless biasing limits are set. Exact methods use standard optimization codes. They require formulating the problem as MINLP or MILP ones. Computation time is also large. Meta-heuristic methods do not guaranty an optimal solution, but can explore very large solution space. They may require expert tuning of their parameters to explore efficiently that space.

The performance criterion associates an objective function built from performance mathematical function, like scaled deviation, Gaussian, sigmoid and desirability. Target type can be below a value, above a value, near a value or within a range of values.

Computer Aided Mixture Design studies are scarce in the literature. They are either dedicated to an application or treated as a set of problems solved sequentially or partially.

In the light of these approaches, we have chosen to implement a genetic algorithm that handles molecular graphs inspired from structure composition graphs and where the performances are calculated through an adapted desirability function. These choices and the method used for their implementation are justified and presented in the chapter 3.

CHAPTER 3

Proposition of a Computer Aided Product Design method

The current environmental and economic issues multiply the number of constraints on chemical products up to a point where finding a single molecule satisfying each constraint is becoming a huge challenge. A way of enlarging the possibilities is to considering a chemical product instead of a single molecule.

In this chapter, an innovative Computer Aided Product Design method adapted to a large number of industrial cases is proposed. In comparison to traditional molecule design or mixture design methods, this method explores a wider solution space, since many optimization parameters exist. The search algorithm is detailed as well as the techniques used for handling the molecule structure and the performance calculation.

3.1 FROM MOLECULE AND MIXTURE DESIGN TO PRODUCT DESIGN

Most of the methods of mixture design seen in the previous chapter only consider the optimization of the ratio between the different specified molecules by changing the mixture composition. The only global approach found (Vaidyanathan and El-Halwagi, 1996) was limited to binary mixtures of polymers because the search method used, namely the so-called exact method, cannot deal with a larger problem.

With product design, our purpose is to go further in the design of mixtures by using inverse formulation and therefore we adapt molecular design method to mixture design. As our project involves also industrial partners and chemists, we have taken into consideration their needs concerning the structure of each molecule in the mixture, and about the necessity to match many property targets. This new approach raises different issues explained in the following paragraphs.

3.1.1 Optimization variables

Going from molecule to product design implies naturally the addition of several optimization variables, concerning the molecules, their composition and also the operating conditions under which the mixture is used as presented in the following figure:

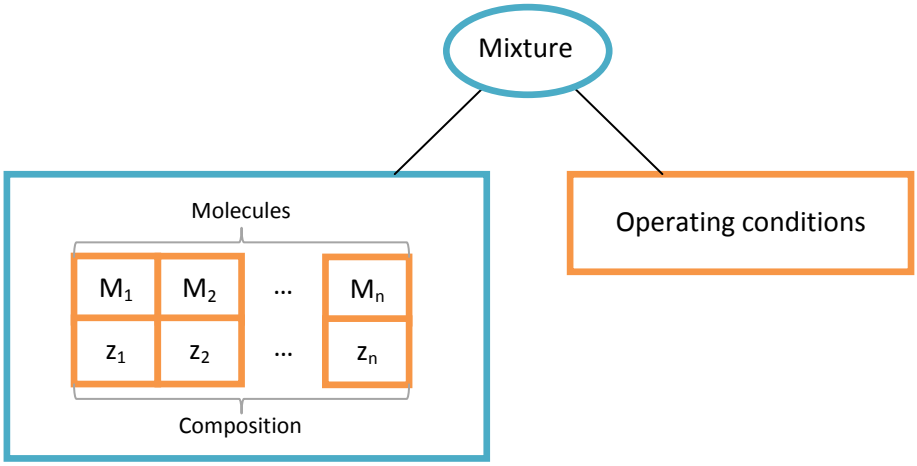


Figure 24: Optimization variables of mixture design

3.1.1.1 Molecules

Within the mixture, the structure of each molecule becomes an optimization variable to be constructed by using traditional CAMD methods from functional groups (basic like OH- or complex like COOH-).

Analyses of industrial problems from the literature have shown that some molecules should be kept untouched within a mixture or search within a database of molecules. For example, within the InBioSynSolv project, we shall look for solvents to solubilize an active ingredient, which structure is known. As seen in section 3.6.1, Conte and Gani (Conte, 2010; Conte and Gani, 2011; Conte et al., 2011) designed sequentially several mixtures where the active ingredient was sought among databases. In another example, Sinha and Achenie (2002) designed a water-organic solvent mixture to wash printing ink. The printing ink did not appear explicitly as a molecule but was rather considered implicitly in the target properties.

To be able to deal with all kind of problem, some of the molecules of the mixture can be:

- fixed (every potential mixture will have this molecule)
- chosen in a list (every potential mixture will have a molecule from this list)
- free being then built from fragments

This is represented in Figure 25.

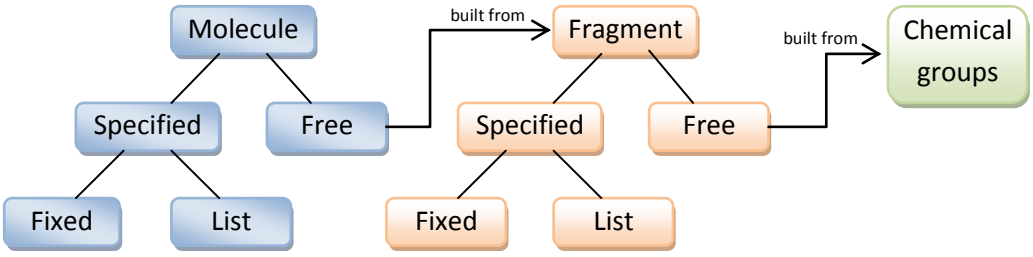


Figure 25: molecular specification possibilities

Notice that in the case where the problem consists in searching a single molecule, this one will also be considered as a mixture, with one molecule and a composition equal to unity.

3.1.1.2 Fragments

With the same logic, it is interesting to constrain parts of a randomly generated molecule. It allows integrating chemical knowledge in the CAMD search, mimicking the trial and error approach of the chemist in his laboratory, used to start from a fixed chemical structure and attempt to functionalize it by chemical reactions. This would allow us exploring the possibilities offered by a renewable source fragment.

As recalled in Figure 25, fragments can be:

- fixed (the fragment will always be present in the molecule)
- chosen in a list (a fragment of the list will always be present in the molecule)
- free being then built from chemical groups.

3.1.1.3 Composition

After making the choices about molecules which can be fixed, we must also consider constraints on the composition value. The optimization of the composition has been well studied in the literature about computer aided mixture design (Sinha and Achenie, 2002; Conte, 2010). Conte et al. (2011) have also considered the search for the suitable composition of a mixture but their scope was limited to find the suitable composition for matching the linear properties, before checking whether the composition matched the nonlinear properties as well. If we take over the example in 3.1.1.1, an active ingredient could have a concentration fixed or, if more flexibility is allowed, a concentration set within a range of values. The method to handle these constraints during the search is detailed later in appendix 10.5.

Instead of setting sequential steps, we shall allow the modification of the composition during the problem solution search, at the same level than a modification of the molecules (Figure 24). The linear and nonlinear properties will be handled equally, rather than sequentially, and will only be considered through the performance evaluation.

3.1.1.4 Operating conditions

The properties of a chemical element depend on the operating conditions (temperature, pressure…). This is particularly true for the physical state: pure phase? (vapor, liquid, solid) phase equilibrium? (vapor – liquid, liquid – liquid, liquid – solid), which can affect in a considerable manner the value of many other chemical properties, like viscosity, heat capacity... So, it is possible that a potentially good mixture is ruled out just because its properties have not been estimated under the right operating conditions. Conte et al. (2011) have considered this issue by implementing a STABILITY algorithm to evaluate the mixture stability in the third sequential step of their MIXD procedure, after searching for the suitable composition matching first linear properties and second and nonlinear properties.

Instead of setting sequential steps, we shall allow the modification of the operating conditions during the problem solution search, at the same level than a modification of the molecules or of the composition (Figure 24).

3.1.2 Property estimation

The estimation of a property for a mixture is estimated with a mixture model and pure compound models.

Regarding the property itself, referring to the classification of properties in section 2.4, we design a CAPD tool that uses properties that can be quantified with a numerical value; implying that any subjective property model will require to be translated into a numerical scale. However if we consider that some people involved in the use of the CAPD tool are not expert enough to choose suitable property methods, we need to make a distinction between real and calculable properties.

3.1.3 Real properties and calculable properties

Inspired by Constantinou et al.(1996) explicit and implicit properties or by Korichi et al. (2008) objective and subjective properties, we define two types of properties:

- the real properties that are understandable by all users but not directly calculable, as for example, volatility, toxicity, fluidity, …
- the calculable properties which are more difficult to understand for a user with limited knowledge in chemical engineering but which can directly be associated to property calculation models. Examples of calculable properties are either temperature dependent properties like vaporization enthalpy, vapor pressure… or temperature independent properties like normal boiling point, critical volume, acentric factor…

A real property will always be associated to one or more calculable properties. Its target will be expressed as an assessment which is translated into computable target values for the associated calculable properties. Consider the example shown in Figure 26.

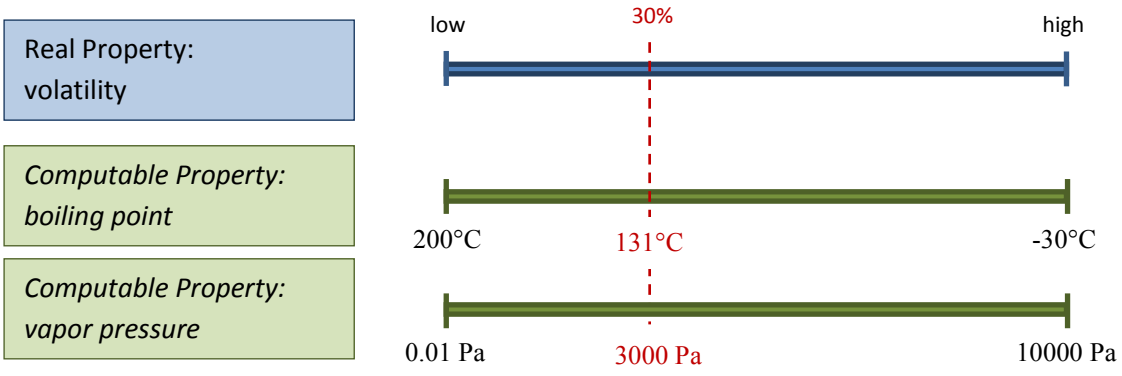


Figure 26: Example of the definition of the target value of a real property

The real property “volatility” is associated to two calculable properties “boiling point” and “vapor pressure”. High and low volatility values are associated to boiling point and vapor pressure values. Thus when the user defines the target volatility thanks to a scale (30%), the corresponding boiling point value is automatically calculated (131°C for boiling point and 3000Pa for vapor pressure) and is set as target of the calculable property. Notice that the scale increases for vapor pressure but decreases for the boiling point.

By setting predefined values for the scales of the calculable properties associated to a real property, the CAMD tool can be used by users other than chemical engineers, either students that have limited time to spend to solve a CAMD problem or business manager… Naturally, the predefined scales should be set by experts in property estimation.

To comply with our choices about property calculation, a calculable property is always associated to a single mixture calculation model and to pure compound calculation models for the concerned molecules in the mixture. These calculation models can change during the search as the level change: more accurate models are used as the level increases. The property is associated to a clear target (such as inferior to 275K).

Overall the calculation of property is described by the following relations.

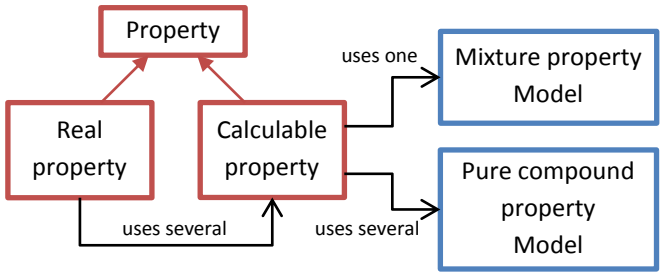


Figure 27: Property management

As illustrated on Figure 27, a property can be either real or calculable. A real property is associated to one or more calculable properties. A calculable property is associated to one mixture property model and to one or more pure compound property models.

3.1.3.1 Mixture property estimation models

A mixture property model estimates the property value of the whole mixture or of a single element within the mixture. The value depends on the composition and the property value of each element taken alone. The linear model and nonlinear models described in section 2.4 have been considered.

Regarding the collection of the compounds in the mixture concerned by the property value, we shall propose to evaluate a property value which only refers (i) to a pure compound or (ii) to a sub-mixture of the mixture considered or (iii) to the whole mixture. Example of (i) is the solubility of the active ingredient in the mixture computed by an solid- liquid equilibrium or the toxicity of one particular compound; example of (ii) is the viscosity of the binary solvent mixture within a four compound mixture; example of (iii) is the boiling temperature of the mixture.

3.1.3.2 Pure compound property estimation models

As seen in the previous section, the estimation of the mixture property requires estimating the property of each element on the mixture by means of a pure compound model.

The most popular predictive models are group contribution methods. In this kind of methods, specific groups of atoms are identified in the molecule and their presence influences the value of the property. A drawback of group contribution methods is that they may fail to estimate the property if some fragments of the molecule are cannot be described by the method list of group. To resolve this problem, we shall optionally propose to use an alternative model if the main model fails to estimate the property of a molecule.

3.2 SEARCH ALGORITHM

3.2.1 Limitation of the previous search algorithms

We have seen in the previous section that the number of variables in our method is much greater than the classic CAMD methods. Subsequently the solution space is much larger. Enumerative methods (Gani et al., 1991) (Gani and Brignole, 1983) (Joback and Stephanopoulos, 1995), which were already challenged by molecular design, are not adapted to such large optimization problems. The addition of new optimization variables is also a problem for exact methods. Indeed, the calculation time may be prohibitive due to the combinatorial explosion.

The meta-heuristic methods are the last methods presented in chapter 2. They are traditionally used for very complex problems as they propose to give a good solution in a reasonable period of computing time. This kind of methods is thus well fitted for a global chemical product mixture design problem that would cover the mixture compounds its composition and operating conditions.

3.2.2 Choice of a meta-heuristic method

The meta-heuristic methods already implemented in CAMD methods are genetic algorithm (Patkar and Venkatasubramanian, 2002), simulated annealing (Marcoulaki and Kokossis, 1998) and Tabu search (Lin et al., 2005).

3.2.2.1 Advantages and drawbacks

The main advantage of meta-heuristic methods is that they will always give a usually acceptable result in a reasonable period of time. But contrary to exact method, this result may not be the optimal result. For CAPD, it is not a severe drawback though. Indeed the property models used to evaluate the mixture have a margin of error. We are thus more interested to have several good mixture candidates to validate in laboratory than to have the best mixture according to calculation models that might not be accurate.

Another problem is that, for the resolution of CAPD problems, these methods are stochastic and hence involves some randomness. Contrary to deterministic method, a stochastic method can be run several times with the exact same conditions and give different results. This can be disturbing for the future users.

3.2.2.2 Genetic Algorithm

We have chosen to implement the genetic algorithm which is a method largely used in several application domains and suits very well the problem considered in the scope of our work. Its use in the context of CAMD is well documented (Patkar and Venkatasubramanian, 2002), including in our team work (Korichi, 2010). Notice that even though we have chosen to implement a genetic algorithm our CAPD method is compatible with other meta-heuristic algorithms like the simulated annealing or the Tabu search.

The principles of the genetic algorithm are based upon the evolution of a population of candidate solutions which favors solutions that best fit the constraints. It consists mainly of three steps as illustrated on Figure 28.

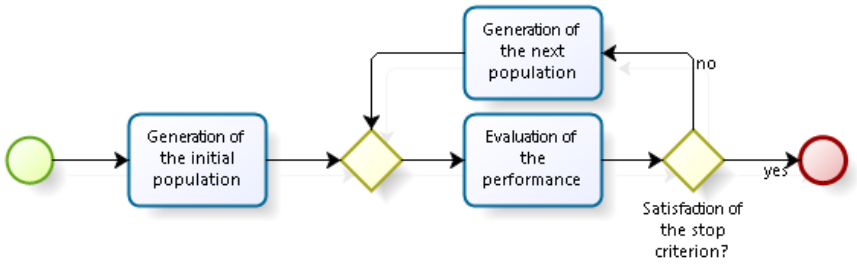


Figure 28: Basic principle of the Genetic Algorithm (BPMN diagram)

In the “generation of the initial population” activity, a population of mixture is randomly created. Then the properties of the mixtures are evaluated and the performance is calculated during the “evaluation of the performance” activity. After that, the performance values are used for the “generation of the next population”. We use the roulette wheel for the selection and genetic operators for the modification as done in the method proposed by Patkar and Venkatasubramanian (2002). The two last activities repeat themselves until a stop criterion is satisfied. The best mixture of the population of the last generation is the result of the genetic algorithm.

The main advantage offered by the genetic algorithm for CAMD is that the population of the last generation contains usually several good mixture candidates which are worth some further testing in laboratory.

3.2.2.3 Elitism policy

The principles of the genetic algorithm through selection and modification lead to a high quality population but do not assure that the best solution generated is in the final result. Indeed, if a solution which perfectly matches the constraints is found, its children will largely be in the next population but the solution itself may be lost. Therefore an elitism policy is necessary. It consists in keeping the best solutions at each population generation. This policy is commonly used and more particularly in CAMD (Patkar and Venkatasubramanian, 2002). It has however some drawback as it amplifies the domination of the best solutions in the search and therefore leads to a less diversified search.

3.2.2.4 Multi-level management

Inspired by the reflections led by Harper et al. (1999) and Korichi et al. (2008), we have transformed the classic CAMD genetic algorithm in a multi leveled genetic algorithm. The main idea is to use at first some simple models when the population size is large and the search space is large, then incrementing the level, to use more complex (and time-consuming) models over a smaller population. This way computational time of complex model calculation is not wasted on poor candidates. The main process is illustrated on Figure 29.

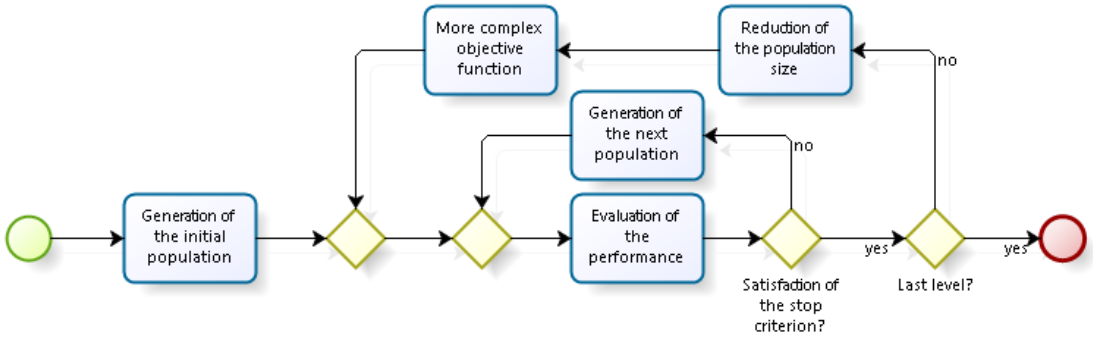


Figure 29: Basic principle of the multi-leveled Genetic Algorithm (BPMN diagram)

A few steps are added, compared to a classic genetic algorithm. When the stop criterion is satisfied the level must be changed. The population size is reduced by eliminating the least fitted candidates and the objective function is modified in order to integrate more complex and more accurate calculation models. Finally, when the stop criterion of the last level is satisfied, the search is finished.

3.3 MOLECULE MANAGEMENT

After having seen which search algorithm we are going to use, we discuss here the choices of the molecular representation model and the operators used to modify the mixtures during the execution of the genetic algorithm.

3.3.1 Molecular representation model

As it is explained in the previous chapter, property estimation models always require molecular information which depends on the molecular representation model used. We have chosen a molecular representation model based on the molecular graphs proposed by Korichi et al. (2008). Once

decomposed into suitable groups, molecular graphs provide inputs to a large variety of property estimation models.

Molecular graphs are also quite easy to understand for the users. One drawback is that modification genetic operators encountered in usual genetic algorithm library packages need to be adapted.

Our molecular graphs consist in matrices where each diagonal element contains a functional group coded as an integer identifier EG with indications on its valency, on its integration in a cyclic structure or not and on its number of hydrogen. Non diagonal integers represent the bonding type (0: no bond, 1: a single bond "-", 2: a double bond "=", 3: a triple bond "≡"). All molecules and fragments in the software representations are encoded this way. The molecular graph of a given free molecule is the aggregation of its fragment graphs completed with the fragment interconnections.

Unlike some linear representation of molecules, either using chemical groups like those used in group contribution methods (Song and Song, 2008) or atomic signature (Weis and Visco, 2010) which can give rise to several isomers for each representation, it describes explicitly the bonds, in a similar manner as other representations based on adjacency matrices (Achenie et al., 2003). An example of our molecular representation model is given in Figure 30.

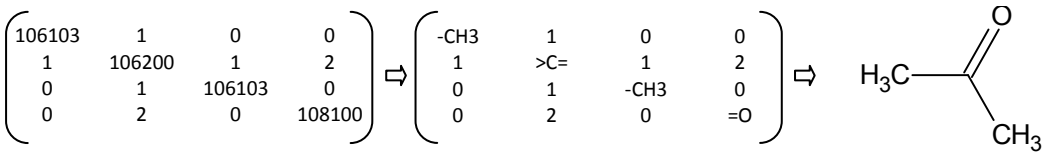


Figure 30: Molecular graph representation of a molecule

The diagonal atomic codes EG = $P_1P_2P_3P_4$ are described in appendix 10.1. P1 refers to the atomic number preceded with a 1 (106 for C, 107 for N, 108 for O, 117 for Cl…), P2 refers to the highest bond order, P3 to the cycle occurrence and P4 to the number of implicit hydrogen bonded on the atom.

The fragments are also represented thanks to graph. However, they need additional information related to their external connections. For example the previous molecule can be managed by the algorithm as two fragments like in Figure 31.

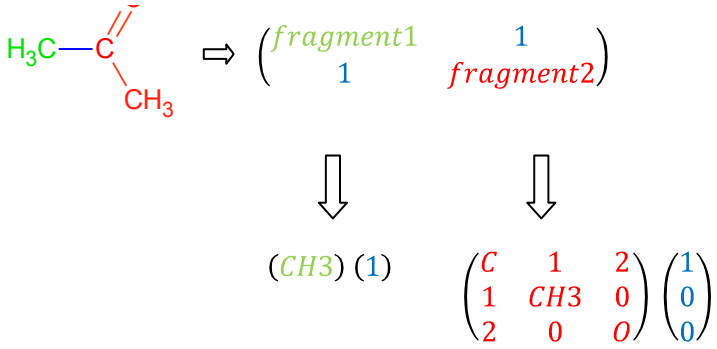


Figure 31: Fragment representation of a molecule

The molecular structure is now be simplified showing just how the two fragments are connected. Both fragments are detailed by their graph and a vector that specifies where and how the fragment is connected.

Further refining is also available, such as describing a whole fragment by a single code. These fragments, named complex groups, are used as any chemical building groups and are distinguished from atomic groups by a specific code starting with 2xxxxxx as in Figure 32. Each code identifies a unique complex group available in a database. Within the InBioSynSolv project, such complex groups have been used to define over 80 biosourced synthons which were them kept as fixed in the molecule structure.

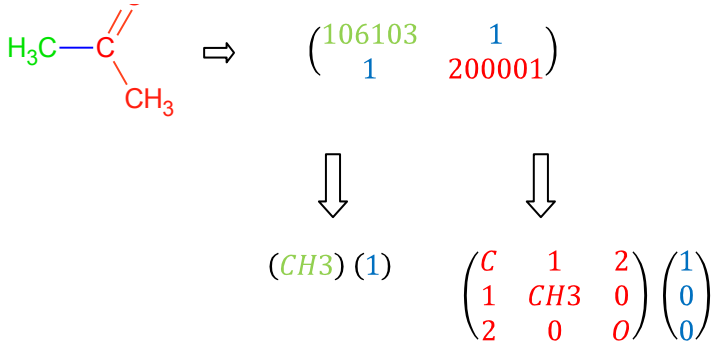


Figure 32: simple and complex group representation of a molecule

3.3.2 Molecule structure modification

As it is done in each CAMD genetic algorithm, modification operators are used. The mutation and crossover operators are classic genetic operators. The insertion and deletion operators are CAMD specific modification operators. All four have been adapted and used by Patkar and Venkatasubramanian (2002) and Van Dyk and Nieuwoudt (2000). In addition, we have introduced a new modification operator

called “substitution” when we realized that the previous four operators could not modify an aromatic cycle without destroying its aromaticity.

All operators are described in details hereunder.

- Mutation

  The mutation operator is a classic genetic operator that is not specific to CAMD. In CAMD, the mutation is adapted to molecular structure and consists in the replacement of a single group by a group that bears the same connections, e.g. >NH by >CH_2 in Figure 7.

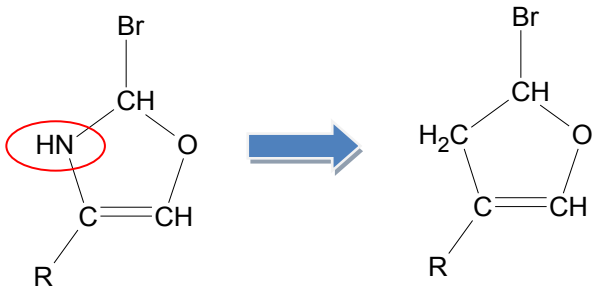


Figure 33: Mutation operator example

The group to be changed and the new group are randomly chosen. This operator is essential to explore the “neighborhood” of a solution.

- Crossover

  The crossover operator is the other classic genetic operator. In CAMD, the crossover involves two molecules.

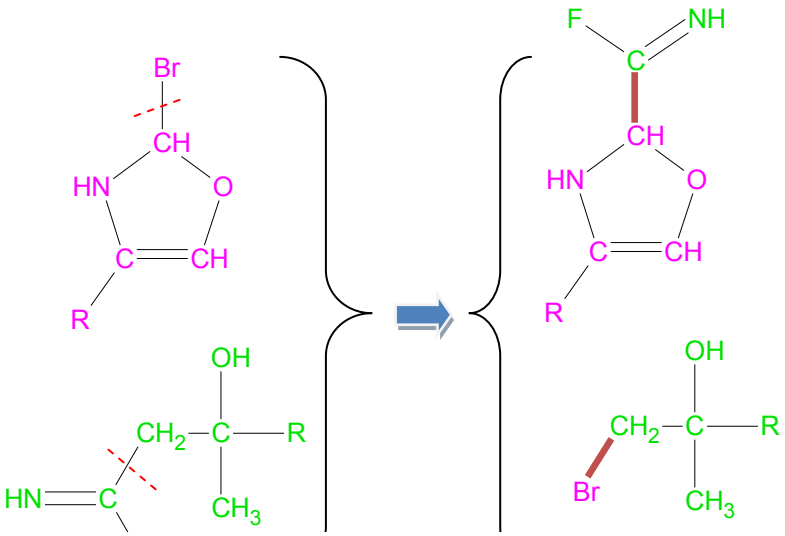


Figure 34: Crossover operator example

A non-cyclic bond of the same type (single, double or triple bound) is randomly chosen in both of the molecular graphs matrices, thus creating four semi-graphs. In Figure 34, the position of the cut (between =C< and >CH_2) is symbolized by a red bold line across the bond. The semi-graphs are then switched and recombined to form two new molecules. Starting with a purple and a green molecule, two half purple – half green molecules are obtained.

The crossover operator allows an effective exploration of the solution space.

- Insertion

  The insertion operator is another usual CAMD operator. It consists in the addition of a group in the graph. We have improved this operator by adding the possibility to insert a group that has more than two connections. This leads to complete the graph with some branches (Figure 35).

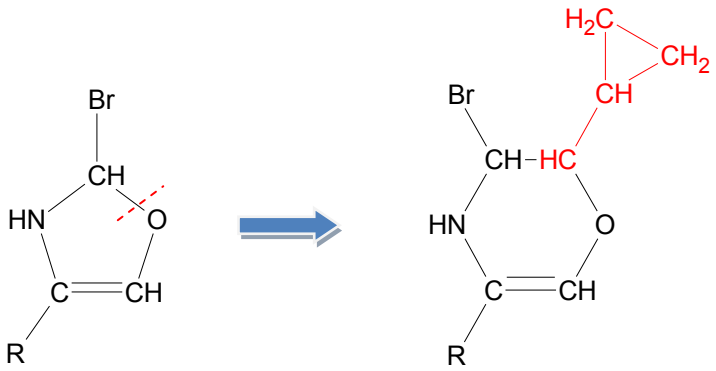


Figure 35: Insertion operator example

A bond is randomly chosen, here symbolized with a red line. Then a group having at least two connections of the type of the bond is randomly chosen (-CH<). If this group has other connection, branches are constructed and added to the graph.

This operator allows exploring the “neighborhood” of a solution.

- Deletion

  The deletion operator is a CAMD operator. It consists in the removal of a group in the graph (Figure 36). To be consistent with the insertion operator we have added the possibility of the deletion of a group that has more than two connections. This can lead to the deletion of whole branches of the molecule.

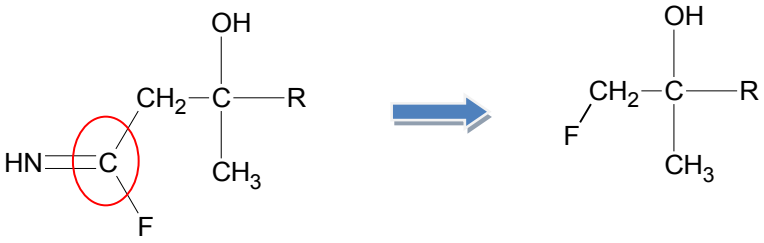


Figure 36: Deletion operation example

A group of the graph is randomly chosen (=C<). If it has at least two connections of the same type, it is a candidate for deletion. The extra branches are deleted and the two remaining branches are directly reconnected. Here the branch NH= is deleted and the group F- is directly connected to the remaining part of the graph.

This operator allows exploring the “neighborhood” of a solution.

- Substitution

  The substitution operator is a new operator. It has been added because the other operator failed to modify aromatic cycles. Indeed the low number of aromatic groups makes it difficult for the mutation operator to be effective. Indeed, the only changes possible are to replace –CH= by –N= and conversely. Then, the crossover operator cannot be applied on cycle. Furthermore, in order not to destroy the aromaticity of the cycle, the insertion and deletion operators can only add or delete the aromatic hetero atoms –O- and –NH- which is insufficient.

  Our substitution operator combines the principles of mutation and insertion. It consists in the replacement of a group by a group that has more connections (Figure 37).

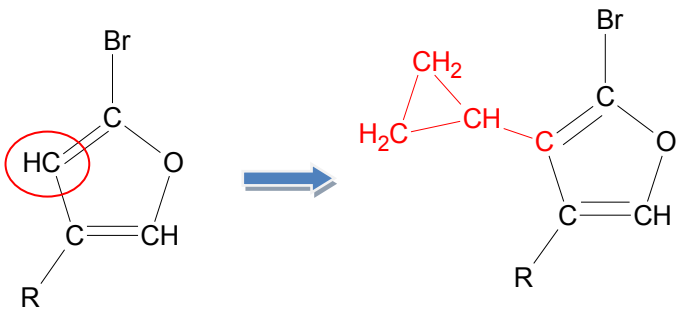


Figure 37: Substitution operator example

Both groups are chosen randomly and the same method as insertion is used to complete the new connections.

3.4 PERFORMANCE CALCULATION

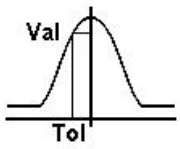
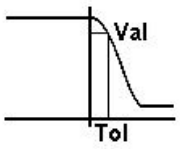
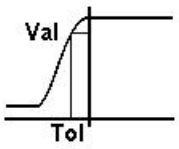
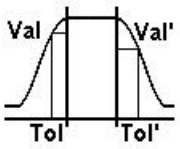
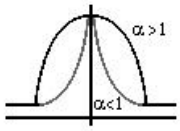
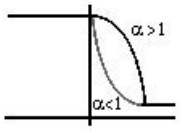
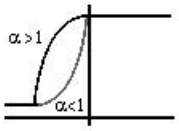
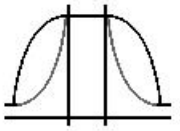
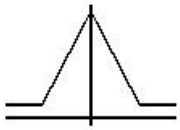
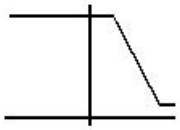
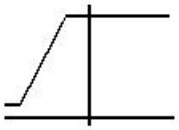
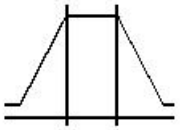
3.4.1 Target values

In order to compare the different candidate solutions, we have chosen to use a performance criterion, which value is in-between 0 and 1. A performance of 0 means that the mixture does not satisfy any of the constraints set on the properties whereas a performance of 1 means that all constraints on the properties are respected.

In order to be able to deal with actual problems, several types of targets must be defined. For example, in the case of a substitution of a molecule, the constraints can be to match a temperature of ebullition (be the closest to a value), to have a toxicity lower than a specified level (be below or above a value) and a viscosity at 298,15K within a range of values (being between two values).

Besides, to allow more flexibility, for each target, we shall propose to the user to choose among several mathematical functions that were detailed in chapter 2: Gaussian-like, desirability-like and straight line like functions. They are recalled in the table below.

Table 4: Performance functions

	Be the closest from a value	Be below a value	Be above a value	Be between two values
Gaussian function				
	$F(x) = G(x) = \exp\left[ln(\text{Val}) * \left(\frac{P-x}{tol}\right)^2\right]$	$F(x) = \begin{cases} 1, & x \le P \\ G(x), & x > P \end{cases}$	$F(x) = \begin{cases} G(x), & x < P \\ 1, & x \ge P \end{cases}$	$F(x) = \begin{cases} G_{inf}(x), & x < P_{inf} \\ 1, & P_{inf} \le x \le P_{sup} \\ G_{\text{sup}}(x), & x > P_{sup} \end{cases}$
Desirability function				
	$F(x) = \begin{cases} 0, & x < P_{min} \\ D_+(x) = \left[\frac{x-P_{min}}{P-P_{min}}\right]^{\alpha}, & x\epsilon[P_{min};P] \\ D_-(x) = \left[\frac{P_{max}-P}{P_{max}-P}\right]^{\alpha}, & x\epsilon[P;P_{max}] \\ 0, & x > P_{max} \end{cases}$	$F(x) = \begin{cases} 1, & x < P \\ D_-(x), & x\epsilon[P;P_{max}] \\ 0, & x > \text{P}_{max} \end{cases}$	$F(x) = \begin{cases} 1, & x < P_{min} \\ D_+(x), & x\epsilon[P_{min};P] \\ 1, & x > P \end{cases}$	$F(x) = \begin{cases} 0, & x < P_{min} \\ D_+(x), & x\epsilon[P_{min};P_1] \\ 1, & x\epsilon]P_1;P_2[\\ D_-(x), & x\epsilon[P_2;P_{max}] \\ 0, & x > P_{max} \end{cases}$
Straight lines				
	$F(x) = \begin{cases} 0, & x < P_{min} \\ L_+(x) = \left[\frac{x-P_{min}}{P-P_{min}}\right], & x\epsilon[P_{min};P] \\ L_-(x) = \left[\frac{P_{max}-P}{P_{max}-P}\right], & x\epsilon[P;P_{max}] \\ 0, & x > P_{max} \end{cases}$	$F(x) = \begin{cases} 1, & x < P \\ L_-(x), & x\epsilon[P;P_{max}] \\ 0, & x > \text{P}_{max} \end{cases}$	$F(x) = \begin{cases} 1, & x < P_{min} \\ L_+(x), & x\epsilon[P_{min};P] \\ 1, & x > P \end{cases}$	$F(x) = \begin{cases} 0, & x < P_{min} \\ L_+(x), & x\epsilon[P_{min};P_1] \\ 1, & x\epsilon]P_1;P_2[\\ L_-(x), & x\epsilon[P_2;P_{max}] \\ 0, & x > P_{max} \end{cases}$

Notice that to our knowledge, the ranged constraint with two half functions (right column in Table 4) was not used in the previous CAMD methods but is useful for industrial cases. Besides, a ranged constraint target can mix two different performance functions. Finally the Gaussian parameters, tolerance and value at tolerance, could be used to account for the model accuracy and confidence respectively.

3.4.2 Objective function

An objective function aggregates all the performance functions of a candidate with respect to each property target into a single performance value so as to be used by the genetic algorithm. This way, the multi objective problem is transformed into a single objective one: maximize the global performance.

There are many ways to aggregate the property performances and we have chosen the simplest one, which is the weighted mean of all the property performances. It can be formulated as:

$$Perf(M) = \frac{\sum_{i=1}^{N} w_i * Perf_i(M)}{\sum_{i=1}^{N} w_i} \quad (21)$$

Where:

- M represents the mixture/molecule considered
- N is the number of objectives on properties
- w_i is the weighting of the i^{th} objective
- $Perf_i$ is the performance of the molecules for the i^{th} objective

The random generation of molecules may lead to chemically unrealistic structures. In order to avoid them, our method penalizes the molecules which do not respect basic feasibility rules such as “no cycle with only three elements”, “no −O−O−O− structure” etc. If one of the0se rules is violated the performance is reduced of a specified percentage. Each rule is associated to a percentage if several rules are violated the highest percentage is used. It can be formalized as:

$$PenalizedPerf(M) = \max_i(\delta_i * V_i) * Perf(M) \quad (22)$$

Where:

- M represents the mixture/molecule considered
- δ_i is equal to 1 if the i^{th} rule is violated, 0 otherwise
- V_i is the percentage to be removed if the i^{th} rule is violated
- $Perf(M)$ is the performance of the mixture calculated with (2)

3.5 CONCLUSION

In this chapter, we presented our proposition of CAPD method. It optimizes composition, mixture structure and operating conditions. A wide range of constraints can be applied to these parameters. For the composition and operating conditions, classical numerical constraints can be set. For the mixture structure, we define three types of molecules; fixed, in a list and free. The free molecules are built from fragments which can themselves be fixed, in a list or free (build with chemical groups). Through the definition of a molecule by its fragments, our method allows exploring the possibilities offered by renewable synthons.

In order to avoid the combinatorial explosion, meta-heuristic methods have been selected, and a genetic algorithm has been chosen for implementation. This search algorithm is multi-level, as proposed by Korichi et al. (2008), and uses an elitism policy.

The molecular representation used is based on molecular graphs with bond type, where basic functional groups and complex functional groups can be handled. The precision of this representation enables to use numerous property estimation models.

The modifications of the molecular graphs are performed thanks to classical genetic operators, mutation and crossover, and CAMD specific operators, insertion, suppression and substitution. Those latter have been improved by adding the possibility of inserting/deleting complete branches.

Regarding the properties, two types have been defined: real and calculable properties. For the evaluation of the performance associated to the properties, several targets can be set: match a value, below/above a value and within a range. With those targets, the performance of each property is calculated thanks to either a Gaussian like, a desirability like or a straight line like function.

The objective function consists in the global performance which aggregates all the performance functions thanks to a weighted mean. If the mixture does not respect some feasibility rules, its global performance is penalized.

A software prototype that implements this method is presented in the next chapter.

CHAPTER 4

Development of the IBSS tool

The Computer Aided Product Design method proposed in the third chapter has led to the development of a software tool prototype called IBSS. This tool is presented here through the software development decisions that have been made as well as through the different steps of the realization: e.g. functional, structural, behavioral and architectural. Then the deployment solutions are detailed and finally validation cases are presented.

4.1 SOFTWARE DEVELOPMENT PROCESS

4.1.1 Software development method

Our development method has been greatly influenced by the IBM Rational Unified Process (RUP) software development method (Kroll and Kruchten, 2003) (IBM, 2011). This process, recommended by the creators of UML (see 4.1.2.1), is an iterative process which is centered on the architecture and is driven by the functional needs. The RUP method is based on best practice principles such as the management of the risks as soon as possible, being sure that the end-user (client) needs are met and making the system adaptable for future changes. The RUP implements these principles in an iterative process. The iterative approach permits the adaptation of the evolution of the requirements. In this way we want to embrace the agility principles as defined by Boucher (2007).

The RUP method has been developed for a specific modeling software application: the Rational Software, but its principles can be used with any other modeling tool. It is specially well suited for large projects involving many people. Since our project is a research and development project aiming for the production of a software prototype with only a few people involved, we have considered that following strictly the RUP process was too heavy for the purpose. Instead we have applied the main principles of the method.

4.1.2 Modeling language

Modeling a system, in our case a software application, simplifies the communication between the different partners of the project and eases the development and the future maintenance. Several standard modeling languages exist. We have chosen to use the Unified Modeling Language (UML) and the Business Process Modeling Notation (BPMN) because they are both well tested modeling languages largely used in the industry.

4.1.2.1 Unified Modeling Language

The modeling language used in our work is UML2 (Unified Modeling Language). This graphical language has been conceived to represent, to specify, to construct and to document the artifacts of systems and specifically software systems. Furthermore, it is a formal and normalized language and highly capable communication support (Booch et al., 2000) (Belaud, 2009). It is an industry-standard language which was originally created by Rational Software, and is now maintained by the computer

industry consortium Object Management Group (OMG). It is the most used modeling language for software design and many tools are available with automation capabilities, such as source code or documentation generation for example.

4.1.2.2 Business Process Modeling Notation

The Business Process Modeling Notation (BPMN) has been developed by the Business Process Management Initiative and the first specifications were released in 2004. Its purpose is the modeling of business processes in a way that is readily understandable by all the business users (White, 2004). It mainly consists of sequences of activities, events and gateways.

In our software development, we used BPMN to model some dynamic aspects of the software.

4.1.3 Software developing tools

4.1.3.1 Programming languages

The software is divided in three independent parts that need coding: the man-machine interface, the search part and the property calculation library.

We first chose Java as programming language because a java application can run on different operating systems (Windows, Linux …). But after one year, we needed to integrate the “MB.dll” Dynamic-Linked Library which calculates the Hansen parameters developed by Professor Steven Abbott and Hiroshi Yamamoto (http://www.hansen-solubility.com). As this DLL is written in C#, a language dedicated to Windows applications, it impaired the platform portability of our software but also caused some advanced interoperability issues with java. Needing the MB.dll features, we decided to restrain the search and calculating parts of the software to Windows systems and to use .NET languages. Finally, the search algorithm is written in C#, the property calculation part is written in VB.NET and the man-machine interface is written in Java.

4.1.3.2 Integrated Development Environment

An Integrated Development Environment (IDE) is a software providing facilities for software development. It combines all the tools necessary, compiler, source code editor, build automation tool and debugger, into one single frame.

The developing tool Visual C# from the Visual Studio 2010 Ultimate suite has been chosen to develop our system. It is edited by Microsoft and is dedicated to the development of C# programs. The Ultimate version allows the developer to construct UML diagrams. Some of these diagrams can then be directly used for source code generation. Inversely a modification of the source code will impact the diagrams. This automation of models alignment between design and implementation enables us to always keep models of our software up-to-date.

4.2 SYSTEM VIEWS

The software application itself is now presented. In the previous section, it has been explained that our software development is model driven. The models resulting from the model driven approach are used in the following sections. But understanding a complex system such as a software application is complicated. UML recommends the use the different views of the system in order to show every aspects of it sequentially. The four views are:

- the structural view
- the architectural view
- the functional view
- the behavioral view.

As represented on Figure 38, the views are organized in two main types. The structural and architectural views are of static types whereas the functional and behavioral views are of dynamic ones. All four views and the associated models are briefly presented in the following sections.

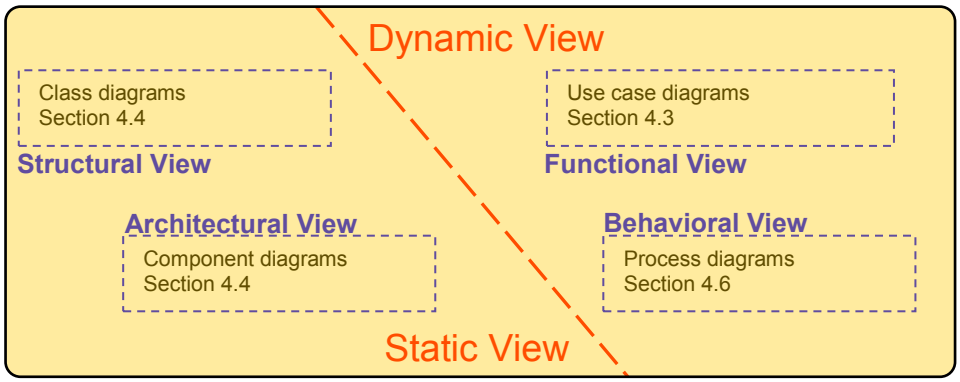


Figure 38: The system views

4.3 THE FUNCTIONAL VIEW

The functional view highlights who are the potential users and which are the main functionalities the software must provide. These functionalities are presented here as a UML use case diagram. This type of diagram shows how the application will meet the needs of the potential users. As it is easy to understand for all partners, even those with no particular competence in software development, the use case diagram is the reference document for the specifications of the application.

For the sake of simplicity, it has been chosen to present only some parts of the use case diagram. The entire use case diagram can be consulted in appendix 10.3.1.

4.3.1 Users

One of the qualities of a software application is to correspond to the need of a large panel of users. In our case we are confronted to the fact that the potential users of our application have contradictory needs. A researcher with expertise in property models for example, will want to have access to as many parameters as possible, in order to get results corresponding perfectly to his problem. On the other hand, a student in the domain of chemistry and chemical engineering will want fast parameters setting in order to get results before the class ends.

We thus have defined two types of users, simple and expert, as represented on the following use case diagram (Figure 39).

Figure 39: the different types of users

The inheritance relationship between the users shows that the expert user (e.g. researcher) will have access to more functionalities than the basic user (e.g. student). A user that identifies himself as an “Expert user” will have access to all the functionalities of the application. A user that identifies himself as a

regular “User” will have access only to the main functionalities. This also allows proposing a simplified version of the application to users with limited knowledge in chemistry.

The application is named IBSS, which corresponds to the initials of the project name. It is designed to offer four main functionalities for the user to launch a search, to create a CAMD problem file, to define properties and models used to evaluate the molecule or mixture property and to evaluate the properties of a set of mixtures. They are now detailed.

4.3.2 Launch a search

The functionality “Launch a search” is the main functionality of the application and corresponds to the CAPD method developed in the previous chapter. It allows the user to search for a product, molecule or mixture, that satisfies a set of properties and constraints that are specified a priori. The main steps of this functionality are represented on Figure 40.

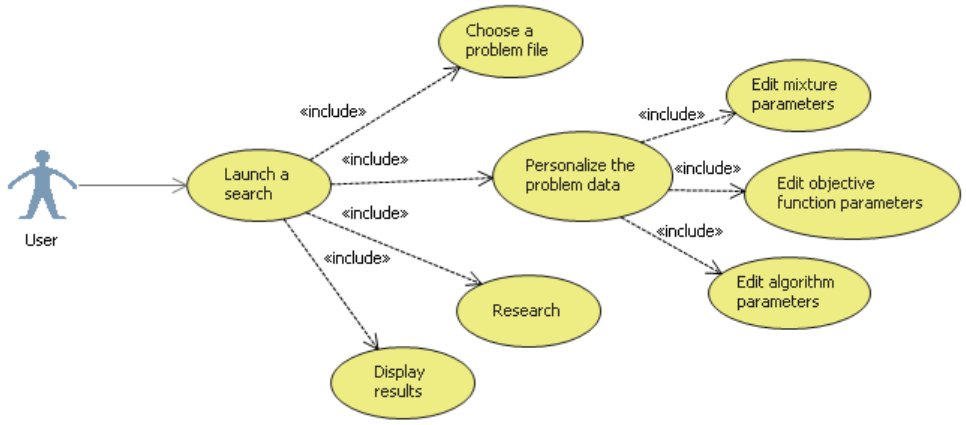


Figure 40: “Launch a search” use case diagram

When the user chooses to launch a search, the application asks him to choose a problem in a list. A problem contains all the data needed for the launching of the search algorithm. The storage and loading of a problem has numerous interests. Indeed, it allows not having to redefine the entire set of data each time a search is launched and it makes it easy for the users with no particular experience in chemistry to use the application, like students.

The user has then the possibility to personalize the data of the problem. Three types of data have been identified:

- The mixture parameters are all the relevant data of the structure of the mixture and of its components. With these parameters, the user can customize the mixture by defining the

possible fixed parts and the degrees of freedom of the different variable parts. The composition and the building blocks of the molecules are found among these parameters.

- The objective function parameters are all the data related to the properties to evaluate, their target values, the property estimation models and the operating conditions used to calculate these properties.
- The search algorithm parameters are all the data that can directly influence the speed and the effectiveness of the search: population size, elitism, etc.

The order in which the parameters are set is preferably the mixture before the objective function as the property calculation models must be chosen for each mixture component which number must be known beforehand.

When the data are correctly defined, the application proposes to the user to launch the search algorithm. The results are displayed as a list of candidate product which he can choose to save.

4.3.3 *Create a CAPD problem and create an objective function*

We have seen in the previous section the interest of the storage and loading of the problem data. For the same reasons it is interesting for a user to be able to create such data sets. These functionalities are only accessible to the expert user as they require some knowledge and experience. Figure 41 shows how the expert user can create a CAPD problem or an objective function.

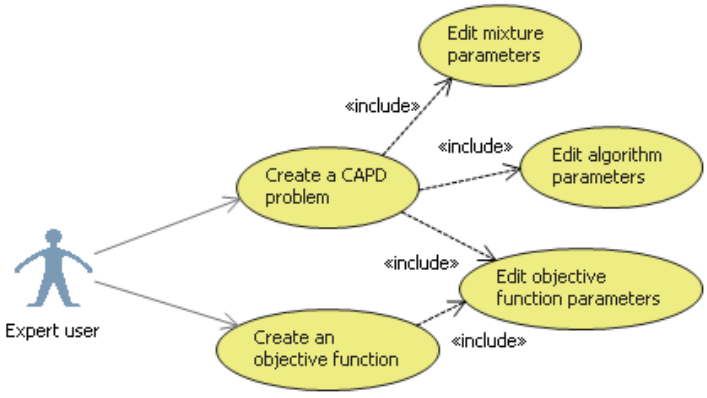


Figure 41: "Create a CAPD problem" and "Create an objective function" use case diagram

To create a CAPD problem, the expert user will define data concerning the mixture, the objective function and the algorithm. To create an objective function, he defines only the data concerning the objective function. The CAPD problem data are then saved into a problem file which can be used to

launch a search. The objective function data are saved as an objective function file that can be used to perform an evaluation or define the objective function parameters within a CAPD problem data.

4.3.4 Create a property and create a model

In order to benefit from the expertise of expert users, our application proposes to create properties and models. Figure 42 shows the associated use case diagram.

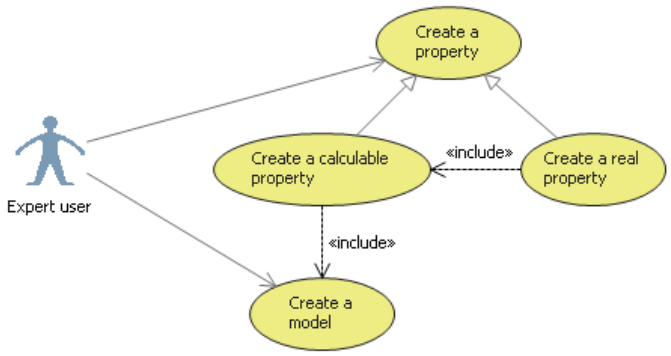


Figure 42: "Create a property" and "Create a model" use case diagram

A calculable property is defined as a property that can be directly evaluated as a numerical value thanks to a property estimation model: boiling point, molecular weight. A real property is defined as a property understandable by all users but not directly calculable (see section 3.1.3). Thus, a real property must be associated to at least one calculable property and a calculable property must be associated to at least one property estimation model.

The user can directly either create a model, a calculable property or a real property.

- To create a model, the user will have to define the input and output parameters and the numerical equation. This functionality is limited to very simple models.
- To create a calculable property, the user will have to choose the models that can estimate this new property among the list of the pre-existing models. For mixtures, a mixture model will first be chosen, either linear or nonlinear. Then a pure compound model will be selected for each compound within the mixture.
- To create a real property, the user will have to choose the calculable properties that can be associated with the new real property among the list of the pre-existing calculable properties.

The Table 5 presents some real properties that are already available in our program and associated with calculable properties and calculation models.

Table 5: Real properties available in our application

Real Property	Calculable Property	Default calculation model
Fluidity	Viscosity	(Conte et al., 2008)
	Molecular weight	
Volatility	Boiling point	(Marrero and Gani, 2001)
	Vapor pressure	(Riedel, 1954)
Toxicity	log(Kow)	(Marrero and Gani, 2002)
	-log(LC50)	(Martin and Young, 2001)
	Log(BFC)	(Veith and Konasewich, 1975)

4.3.5 Evaluate mixtures

Taking advantage of the ability of the CAMD tool to evaluate mixtures during a search, the user can evaluate any mixture independently from a search, with respect to an objective function. As the application can evaluate mixtures during a search, we thought it would be interesting to offer this functionality directly to the user. This way the user can for example evaluate his products and compare them to search results. Figure 43 shows the different steps of the mixtures evaluation functionality.

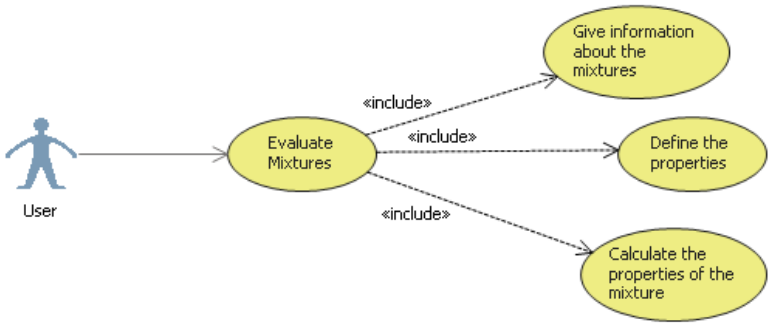


Figure 43: "Evaluate mixtures" use case diagram

The user first gives the information about the mixtures, such as SMILES description of the molecules and the composition. Then he defines the properties and the property estimation models. All properties are aggregated in an objective function and some targets can be defined if the user wants to rate the mixtures. Both evaluation mixture data and objective function data are saved as data files and can be reused for another evaluation.

A restriction exists: the properties are defined for a specific number of elements in a mixture. It is thus important that all the mixtures have that same number of elements and a systematic check is performed before the evaluation.

4.4 THE ARCHITECTURAL VIEW

The architectural view presents the different software components that constitute the IBSS application.

4.4.1 Overview

We have chosen to divide the application into three components. Each component is independent and can easily be reused in other application.

The three independent components are the man-machine interface (MMI), the search and the property calculation component. The search and the property calculation components form the CAMD program (Figure 44).

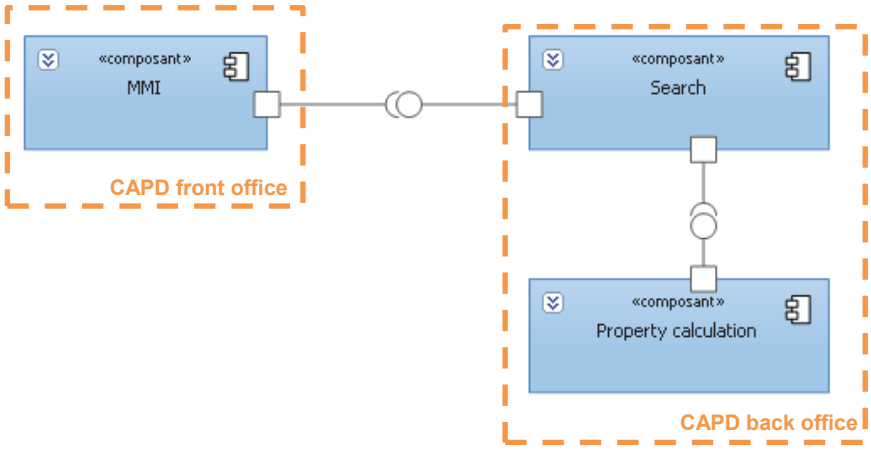


Figure 44: Component Diagram

The three components and their interface of the IBSS solution are presented in the following sections.

4.4.2 MMI component

The man-machine interface is independent from the CAMD program. The user uses the interface to set the parameters of his problem. The parameters are then written in an XML file that will be set as input of the CAPD program. This way the user does not need the program to be available or installed on his

computer to create a problem. The user can launch the CAPD program via the interface by providing the name of the XML file containing the parameters value and the name of the exit files.

4.4.3 Search component

The search component manages the search algorithm. It can be launched directly or via the MMI component. It initializes itself with the user information that is contained in the input XML file. It generates mixture candidates and modifies them to investigate the solution space. The evaluation of the properties of all the potential solutions is performed thanks to the property calculation component. The search component then uses the property values to calculate the performance of each candidate solution with respect to the target values set in the objective function.

4.4.4 Property calculation component

The Property calculation component is a Dynamic-Link Library (DLL) called MolPropEstim.dll. It already proposes numerous property calculation models listed in appendix 10.2 and will be continuously updated in order to offer our application up-to-date models and solve problems over a wider range of application domain. The interface of the DLL is represented on Figure 45 with its main attributes and methods.

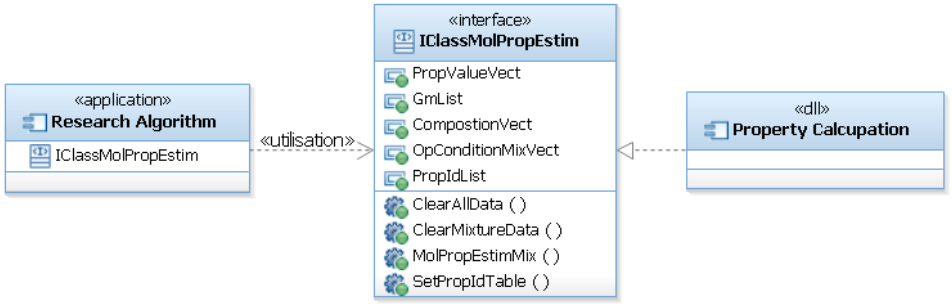


Figure 45: Interface of Property calculation service (MolPropEstim.dll)

During the search, when the ObjectiveFunction object is created, the set of the property estimation models to compute is transmitted by the search component thanks to the “PropIdList” attribute. With the “SetPropIdTable” method, the property calculation component elaborates a calculation sequence which determines the order in which the different models and sub-models are calculated. This prevents the duplications of calculation processes and thus saves computational time.

At each mixture evaluation, the search component pushes forwards all the information of the mixtures: molecular graphs (via “GmList”), composition (via “CompositionVect”), and operating conditions (via “OpConditionMixVect”). Then it orders the evaluation by calling the “MolPropEstimMix” method (Figure 45). The property calculation component launches the calculation sequences. It may happen that a model specified by the user is not adapted to the present mixture. In such cases, the calculation component can decide, if authorized by the user through an attribute within the “PropIdList”, to substitute the chosen model by a different model that is better suited.

The property values, “PropValueVect”, are then sent back to the search component which uses them to calculate the mixture performance.

These three components are developed in parallel in the frame of our project and different persons are in charge of their implementation. Our focus in this thesis is the search component.

4.5 THE STRUCTURAL VIEW

The structural view presents the modeling abstractions (the classes and the relationships that exist between them). Here, only the structural aspects of the implementation of the search component are presented, thanks to package and class diagrams.

4.5.1 UML packages overview

We have identified four distinct parts; each file containing any source code has been assigned to one of these parts. These parts, called packages in UML, are presented in the following paragraphs.

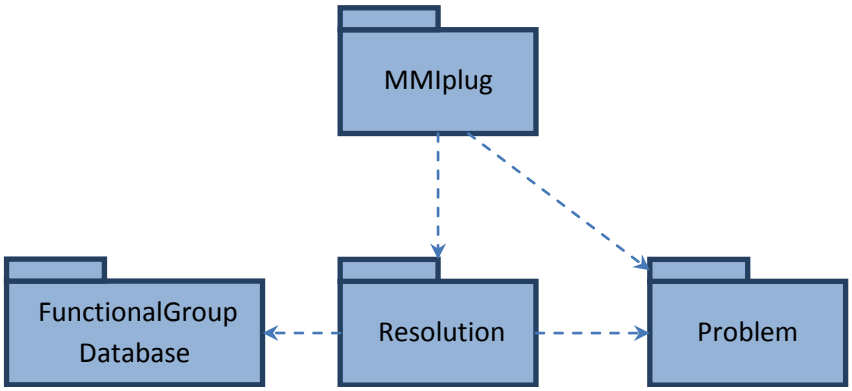


Figure 46: Package diagram

The MMI plug package uses both the Problem and the Resolution packages. The Resolution package uses the Problem package as well and the FunctionalGroupDatabase package.

4.5.2 The MMI plug package

The interface of the application allows a user (man or machine) to use the method. Attention must be paid to the fact that the interface discussed here is not the MMI component presented in 4.4.2, but an interface integrated in the search component. It is constituted of 5 classes interacting the way shown on Figure 47.

Figure 47: The MMI plug package

The Program class is the main class of the application. It is the one called when the application is started. If the application is launched by another application an object Interface is created. Otherwise, if the application is directly launched by a user (developer) a UserInterface object is created. Thanks to the inheritance relationship, this last object has the same behavior as an Interface object but it will also ask the user what he wants to do. Both interfaces use the BackupObject methods to load or save problem data in XML format. They use the ExitFileWritter methods to write the results files. An Interface object manages the Problem and the Resolution packages.

4.5.3 The Problem package

The Problem package contains all the data of a search to launch. It is totally uncoupled from the resolution procedure for two main reasons. First it allows a clear XML serialization and deserialization (transcription of the data into an XML file and inversely). This way loading and saving data is simple. Second, it allows organizing the data in an understandable manner for the user, instead of an organization optimized for computing like what is done in the Resolution package.

Thus the Problem and Resolution packages are very similar. They have distinct classes but some shared name. In order to avoid any confusion, the classes from the Problem package have a name starting with “Pb”.

We describe here step by step the class diagram that represents how the data is stored. The whole diagram is given in appendix 10.3.2.

The Figure 48 shows the main classes of the problem package.

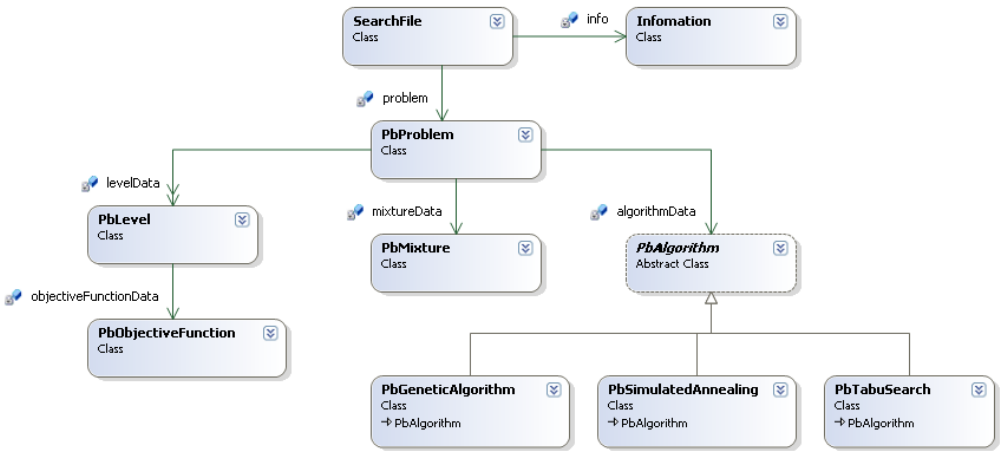


Figure 48: Root of the Problem package

It can be seen on Figure 48 that the problem package structure respects the organization of the problem data introduced in 4.3.2: the data on the algorithm, on the objective function and on the mixture is separated.

The search algorithm can be chosen among several meta-heuristic methods but for the moment only the genetic algorithm is implemented. In addition to the search algorithm, a problem object is also associated to objective functions via level objects and to a mixture data object.

4.5.3.1 The objective function structure

Figure 49 details the objective function data structure in the Problem package.

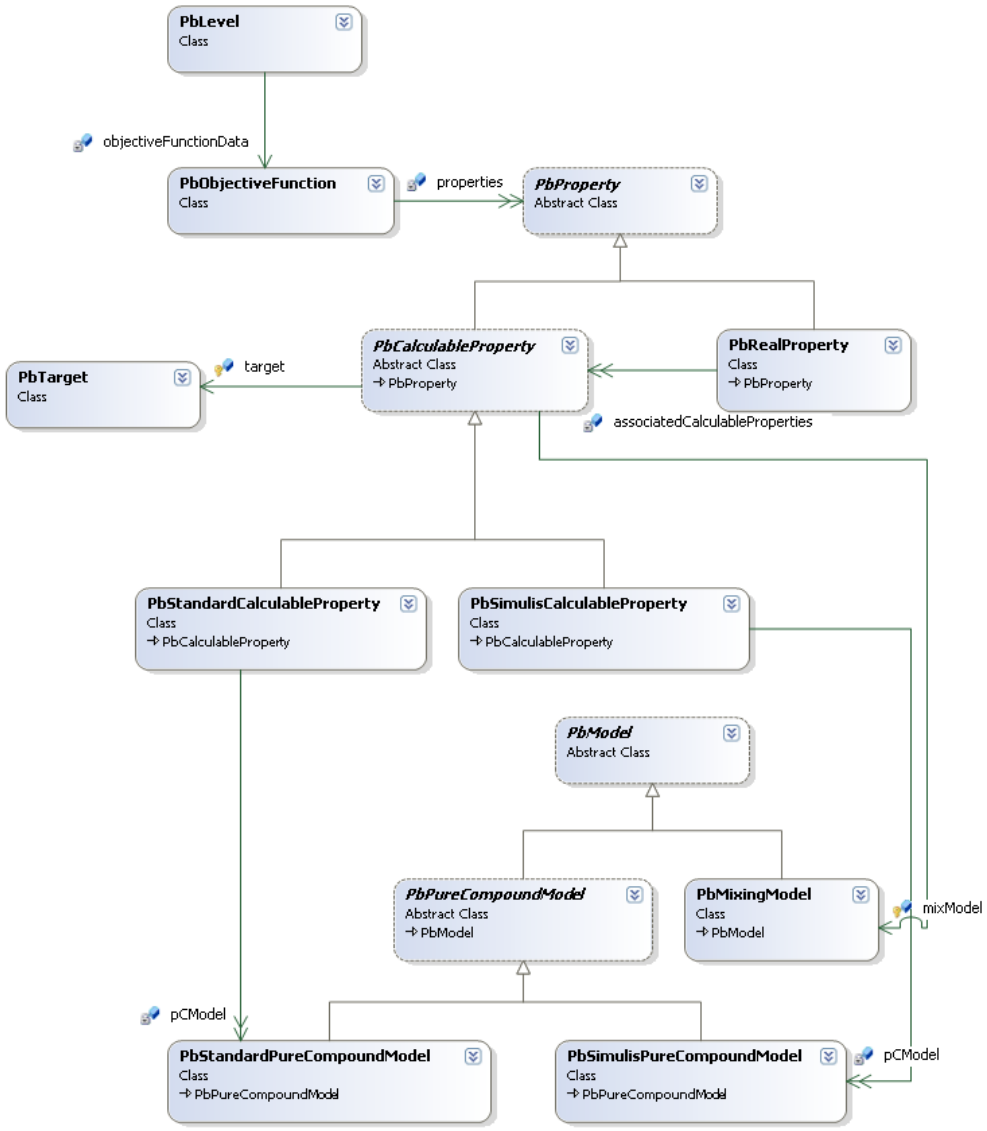


Figure 49: Structure of the objective function data

As represented on Figure 49, a different objective function is associated to each level. This way the user can, at each level, add properties or change property estimation models to more complex ones in the spirit of the multilevel approach described in chapter 3 (section 3.2.2.4). The objective function aggregates properties that can be either real or calculable as formulated in chapter 3. Each real property is associated to one or more calculable property as was illustrated in section 4.3.4. Each calculable property is associated to a mixture model and to several pure compound models. The Simulis properties are calculated with the SimulisThermodynamics.COM component program (Baudouin et al., 2008) which bears its own interface. For this reason, the process of collecting the data is not the same as with a standard calculable property, and they shall therefore be differentiated. Each calculable property is associated to a target object.

Figure 50 shows how the targets are defined.

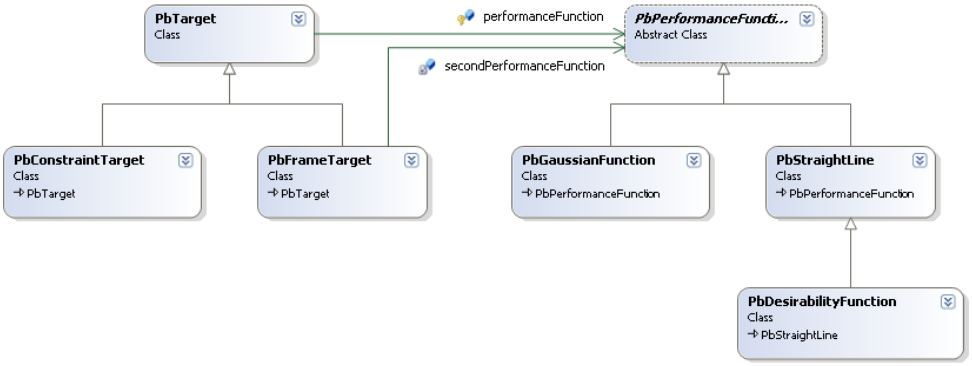


Figure 50: Structure of the target data

In accordance with what has been said in chapter 3, we have three types of targets:

- An equality target, “PbTarget”, deals with constraints where a property must be the closest to a value.
- An inequality target, “PbConstraintTarget”, is a constraint where a property must be below or above a value.
- A frame target is a constraint where the property value must be within a specific range.

Each type of target is associated with a performance function, except the frame target which is associated with two. The three types of performance functions seen in the previous chapter are available, but any other function can be implemented.

4.5.3.2 The mixture structure

Figure 51 details the structure of the mixture data in the problem package. The structure is complex in order to respect all the specifications made in chapter 3.

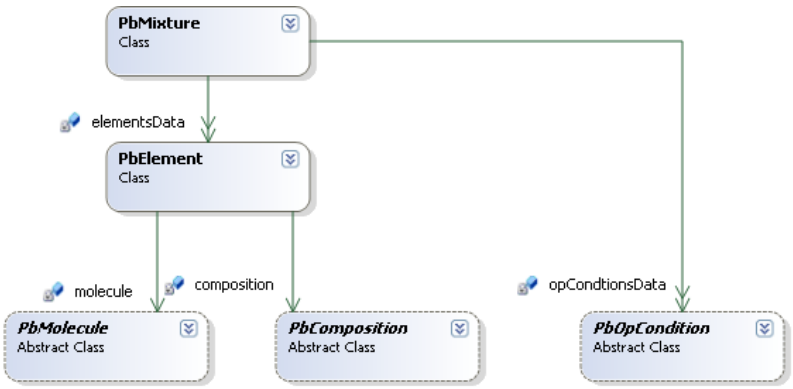


Figure 51: Structure of mixture data

As specified by Figure 51, a mixture is made of elements and operating conditions. Each element of the mixture is associated with a molecule and a composition value.

Regarding composition and according to the needs recalled in chapter 3, we have three types of composition as represented in Figure 52.

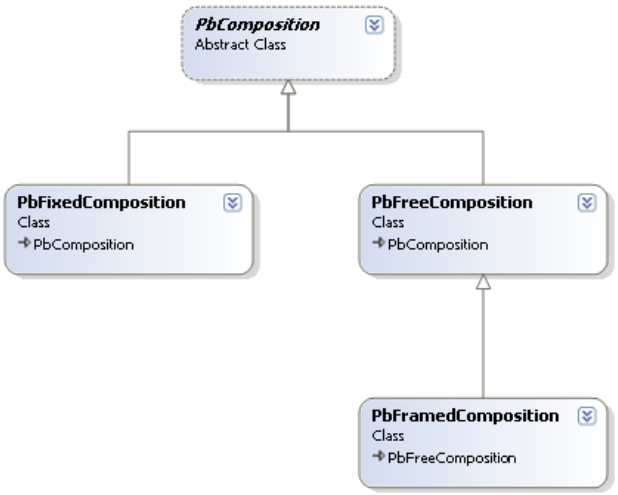


Figure 52: Structure of the composition data of an element in the mixture

- For the elements with a fixed composition, a value between 0 and 1 is given. This value will be the composition value of the element during the whole search.
- For the elements with free composition, no value needs to be given. The composition value of those elements will vary during the search between 0 and 1 while respecting the need for the sum of the composition over all elements to be equal to 1.
- For the elements with a framed composition, a minimal and a maximal value, both between 0 and 1, are set, defining the range where the composition can vary during the search. The composition value of those elements will vary during the search but will always be between the two defined values. The inheritance relationship means that a framed composition is a special free composition. Here, it has two additional attributes: the minimal and maximal value.

Regarding the molecule of each element of the mixture and with a logic similar to composition, some of the molecules of the mixture could be fixed (every mixture candidates will have this molecule) or could be chosen in a list (every mixture candidates will have a molecule from this list), or could be free, then being built from fragments. Our structure corresponds to these requirements as shown on Figure 53.

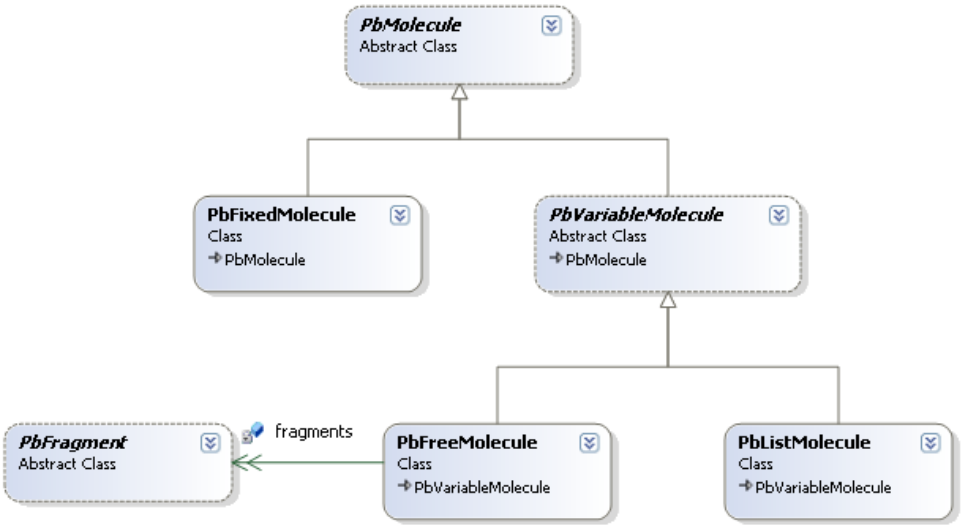


Figure 53: Structure of molecule data

- A fixed molecule will not change during the search. Its description is given by the user.
- A list molecule can only be chosen in a list of molecules given by the user. During the search, if this element is asked to change its molecule value, it will choose another value in the same list.
- A free molecule is defined by some fragments connected according to a flexible structure.

Figure 54 shows an example where a molecule is made of four fragments.

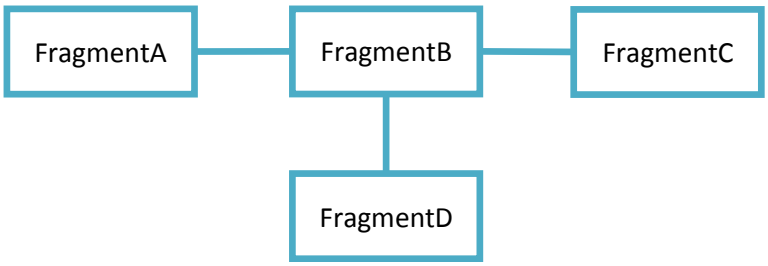


Figure 54: Illustration of a free molecule

The structure of the molecule remains the same all along the search. In other words the different fragments may be modified but they are always connected together the same way. The only restriction to this representation is that two fragments cannot be connected with two different bond types.

Keeping the same logic, the specifications given on fragments in chapter 3 have led to the following structure (Figure 55). Three different types of molecular fragments can be defined.

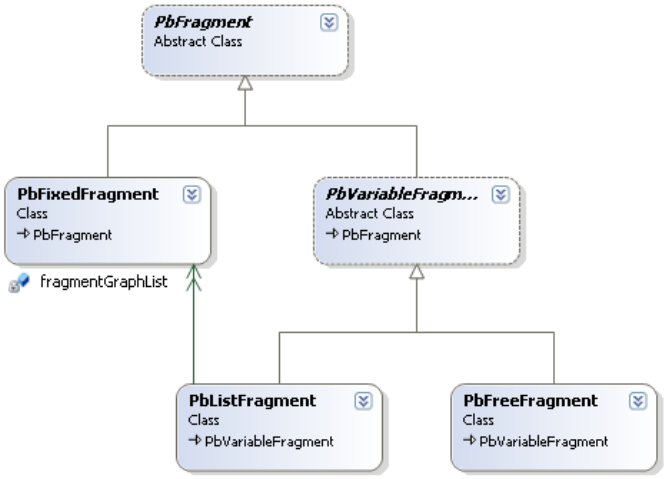


Figure 55: Structure of molecular fragment data

- A molecule that contains a fixed fragment will always contain this fragment. Its description is given by the user.
- A molecule with list fragment will always contain a fragment belonging to a list given by the user. During the search, if the molecule is asked to change this fragment value, it will choose another value in the same list. There is one restriction: the fragments in the list must have the same bond number and type.
- A free fragment is built by assembling different chemical building blocks that are selected by the user. During the search these kind of fragment can be modified by the modification operators described in the previous chapter.

Regarding the operating conditions data of a mixture, Figure 56 displays their structure.

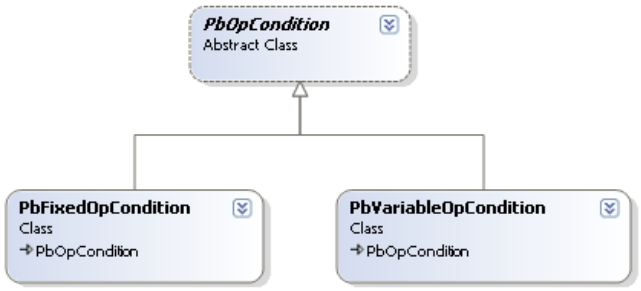


Figure 56: Structure of the operating condition data

Operating conditions can be either fixed or variable, then enabling to find the best operating conditions for the new mixture, as it was explained in chapter 3.

4.5.4 The Resolution package

The Resolution package is the dynamic and central part of the application. It coordinates the entire search algorithm, including mixture handling and performance calculation. It initializes itself with the user information that is contained in the Problem package. It uses the information of the functional groups thanks to the FunctionalGroupDataBase package and calls the property calculation DLL for the estimation of the property calculation models.

The Figure 57 shows the main classes of the resolution package.

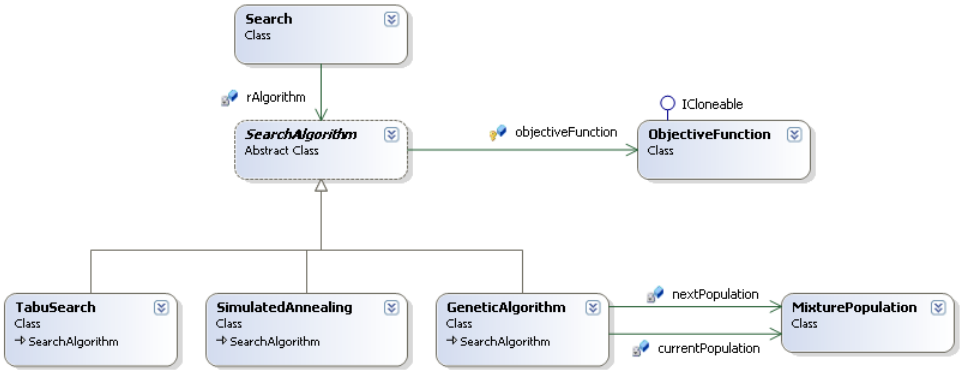


Figure 57: Root of the Resolution package

When a search is launched, the interface (MMI package) creates a search object which launches the search algorithm specified in the problem package. As shown in Figure 57, each search algorithm is associated to an objective function, and the genetic algorithm is additionally associated to two mixture populations: the current population which is the population of the “parents” and the next population which is the population of the “children”.

Compared to the Problem package, there is only one objective function in the Resolution package. When the level changes, the current Objective function is deleted and the next one is loaded with the properties constraints specified in the Problem package. The structure of an objective function in the resolution package and the one in the Problem package are similar.

The Figure 58 describes the mixture population structure.

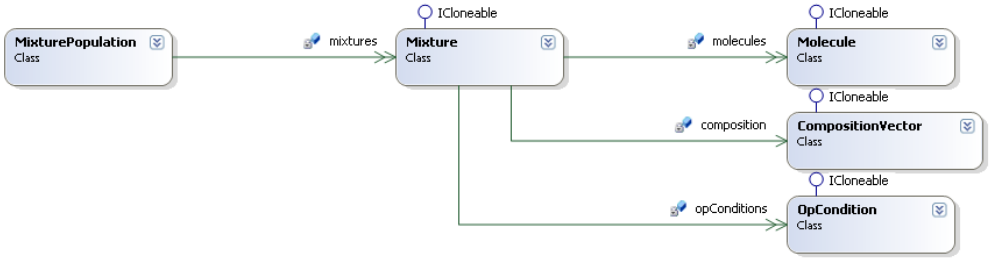


Figure 58: Structure of a mixture population

As shown on Figure 58, a mixture population is made of several mixtures. The structure of a mixture is very similar in the Resolution and in the Problem package, except that here the composition is managed as a vector. The element by element definition, used in the Problem package, is convenient for the user but not for a computer which would rather access composition data in a vector, for example to make sure that the sum of the composition is equal to 1.

Regarding molecular fragments, the architecture in the Resolution and in the Problem packages is very similar except for the FreeFragment class which uses two other classes in the Resolution package as illustrated by Figure 59.

Figure 59: GraphConstructor and GraphModifier class

The GraphConstructor and GraphModifier are classes that respectively construct and modify graphs. They are used by every FreeFragment object when being created or when a modification operator is needed. They both use the FunctionalGroupDatabase package to access the information about the functional group, such as the type and the number of connection of the group.

These two classes are the classes of the application where the largest coding effort was needed during the development process evolving from handling simple chemical structures to most complex structures. First, the application was only able to create and modify linear and branched molecular structures. Then the management of the external connections has been added, enabling a fragment management, finally, cycles, then fused cycles and finally aromatic cycles managements were implemented, each adding complex chemical building rules.

Note: during the course of the development, new methods were added incrementally to these classes. Step by step, the code has become too large with more than 3000 lines of source code for each class. Further, the structure of the code is too procedural to be a real object architecture. For the sake of maintenance, as future evolution of the IBSS, we recommend to reengineer these classes into an independent reusable component.

4.5.5 The FunctionalGroupDataBase Package

The FunctionalGroupDataBase package contains all the data of the functional groups used to build free fragments. The class diagram given in Figure 60 is more detailed than the previous ones. Indeed as the data of functional groups is important, the attributes of the classes are presented here.

Figure 60: Class diagram representing the different types of functional groups

All the chemical building blocks are accessible via the FunctionalGroupsSetDatabase. They can be basic or complex functional groups.

Basics functional groups (e.g. $-CH_3$, $-OH$…) are instances of the FunctionalGroup class. Being single non-hydrogen atom (S, C, N, O…), they allow creating a great diversity of molecules and are useful in many property estimation models like group contribution models which identify groups from one or more basic functional groups. In a molecular graph, they are encoded as an integer as proposed by Korichi et al. (2008) and follow the formalism detailed in appendix 10.1. This integer is unique for each group and is stored in the “id” attribute. For the molecule creation and modification, more information is needed. It is stored in the following attributes:

- "bondVector", a vector representing the number of each type of connection of the group. With this attribute the number and type of the bonds of each group is known, which is essential for the creation and the modification of molecular graphs.
- "canBeAromatic", a Boolean value that is true if the group can be part of an aromatic cycle. This attribute allows the algorithm to manage aromatic cycle.
- "id", an integer that identifies the group and that is understandable for the algorithm.
- "userId", a string that identifies the group and that is understandable for the user. For $-CH_3$, it would be "CH3−". It allows the user to quickly understand the graph.

Complex functional groups are instances of the ComplexFunctionalGroup class. Contrary to basic groups, they are multi-atomic groups. They allow directing the search toward molecules coming from sustainable sources as the user can select complex groups that represent chemical structures specific to a given raw material. Examples are C5 and C6 sugar ring, glycerol backbone, fatty acid …

The inheritance relationship with the FunctionalGroup class means that all the attributes from the FunctionalGroup class are attributes of the ComplexFunctionalGroup class.

A complex functional group needs supplementary attributes which are:

- "graph" which is the representation of the complex group in terms of basic groups.
- "connection" which is a vector containing a list of integer and which represents the location of the connections of the group in the "graph".
- "strConnection" and "strGraph", strings that contains the data of "connection" and "graph" in an XML compatible format.

Basic and complex group can be easily differentiated thanks to their "id" attribute first digit: a basic group "id" will always start with a 1, whereas a complex group "id" will start with a different digit, 2 for example. The "id" attribute of the group allows the software to access to all the attributes of the functional group, and in particular to "graph" and "connection" attributes of a complex group as represented on Figure 61.

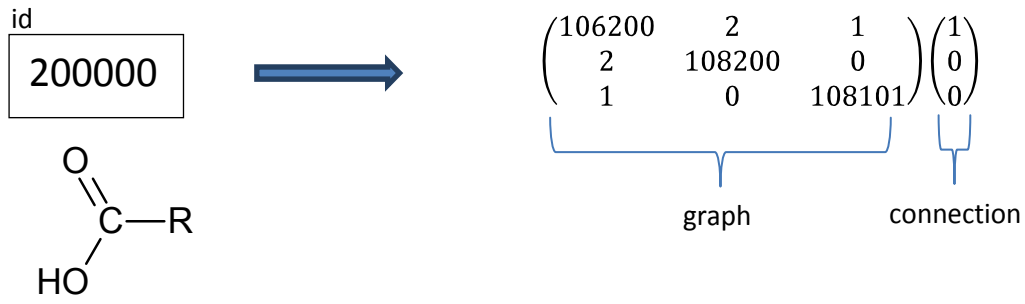


Figure 61: Information associated to a complex group

These attributes are essential as the property estimation models usually need information provided by the basic functional groups. The molecule/fragment structure will thus be written with two different types of graph:

- The complex graph which contains both basic and complex groups. It is constructed by the "create graph" method and is used for all the modifications of the molecule/fragment in order to guaranty that the complex groups remain untouched.
- The expanded graph which is the exact translation of the complex graph into a graph which contains exclusively basic groups. It is used for the evaluation of the molecule properties.

The Figure 62 represents those two types of graph for the propionic acid, here described by a complex carboxylic acid functional group and two basic groups $>CH_2$ and $-CH_3$.

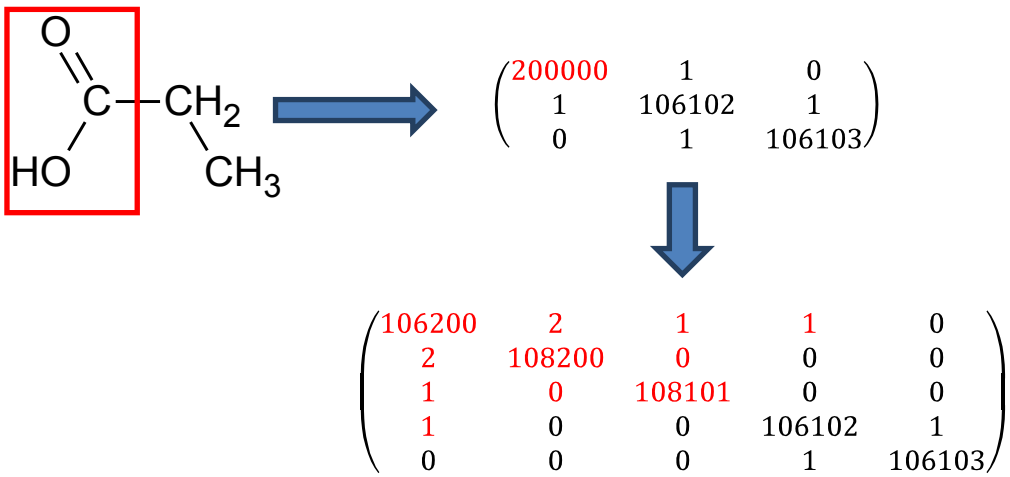


Figure 62: Example of a complex graph and an expanded graph

4.6 THE BEHAVIORAL VIEW

The behavioral view presents how the different processes are implemented in the software in a specific situation and for a given result. The sequencing of the activities is presented thanks to BPMN diagrams and UML activity diagrams.

4.6.1 The behavioral view outline

The global overview of the different activities during a search is presented in Figure 63. Some of these activities are detailed in the following sections. This overview also shows how the different components and packages are working together.

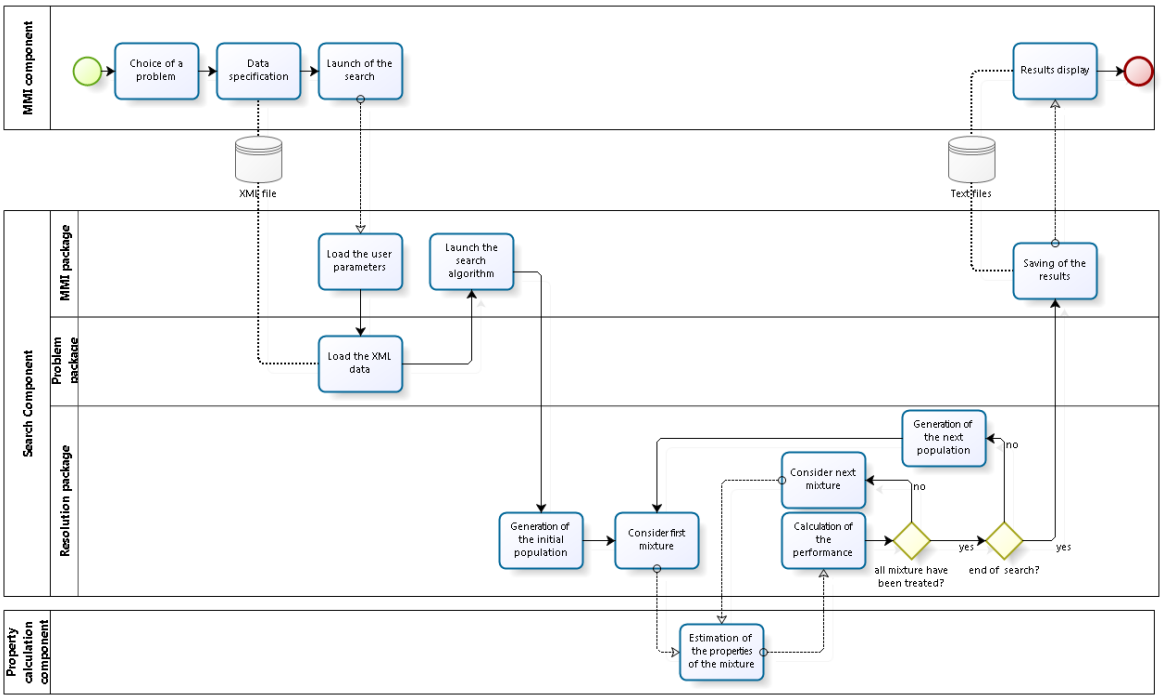


Figure 63: BPMN diagram giving a global overview of the system behavior

First via the Java MMI component, the user chooses a problem to be solved, that is to say the parameters of the problem he wishes to solve. He can edit them afterwards (more details in 4.6.2) and save them in a specific XML file. He then launches the search. The MMI package of the search component takes over by ordering the loading of the parameters. In the Problem package, a problem object is created. It contains all the data in the XML file. Then the search algorithm (here a genetic algorithm with a single level) is launched: the Resolution package generates the initial population (more details in 4.6.3 and 4.6.4). Then all the properties of each mixture of the population are evaluated by the property calculation component. These values are used to calculate the performance of each mixture.

Once the performance of every mixture is evaluated, the next population is generated (more details in 4.6.5). This new population is then evaluated. This sequence goes on until a stop criterion is satisfied and the search is over. The results are then saved in a text format be the MMI package of the search component. The data in these files are afterwards displayed by the MMI component.

The activities of data specification, mixture creation, fragment creation and mixture modification are detailed in the following sections.

4.6.2 Data specification

A process of specification of the parameters is proposed to the user. Indeed interdependences exist between the different parameters of a problem. For example, the properties of the objective function are dependent of the number of elements in the mixture. It is thus better to set some parameters before others. Defining this process allows finding the sequencing where the interdependences will not affect already specified parameters.

Four steps have been identified in this process:

1. The specification of the mixture structure
2. The specification of the objective function
3. The specification of the operating conditions
4. The specification of the algorithm parameters

These macro-activities are sequenced as illustrated in Figure 64.

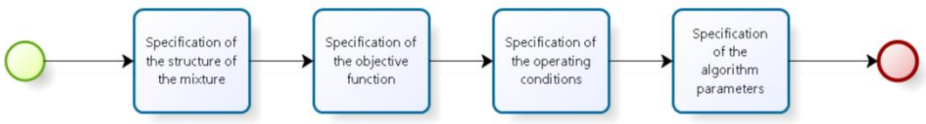


Figure 64: Macro-process of the specification of a problem data

During the first activity, the user specifies the constraints on the molecules and on the composition. Then he defines the objective function using the number of compounds in the mixture to properly set the property estimation models. The third activity consists in finishing the setting of the parameters of the mixture by defining the constraints on the operating conditions of the mixture. This step cannot be performed earlier, within the first step for example, because the operating conditions, that are required, depend on the property estimation models that are going to be used. This step thus must be performed

once the objective function is fully defined. Finally the parameters of the genetic algorithm are set. This step is put at the end of the process as all the mixtures parameters, molecules, composition and operating conditions must be defined before specifying the algorithm modification probabilities.

4.6.3 Mixture creation

A method for creating a random set of mixtures is used for the generation of the initial population. All the optimization variables seen in the previous chapter, namely composition, the molecular structures and the operating conditions of the mixture, are set randomly in order to offer a high diversity of the mixtures in the initial population. This diversity is necessary for the genetic algorithm as it guaranties a better investigation of the solution space. The process that has been implemented in our application is illustrated on Figure 65.

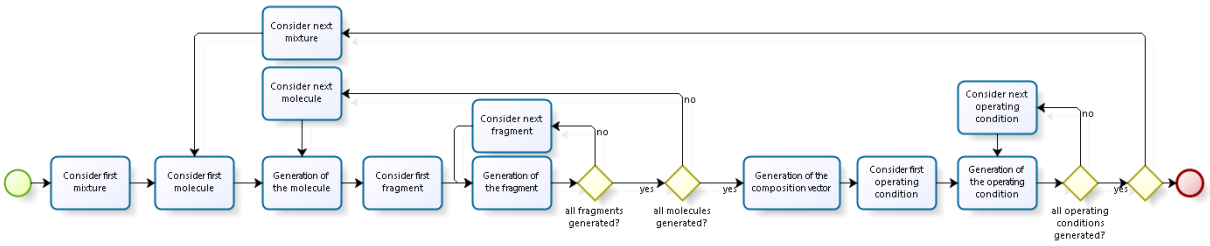


Figure 65: Simplified process of a mixture creation

- The mixture creation starts with the generation of all the molecules. If the molecule is fixed, nothing is to be done. If the molecule is a list molecule, a molecular graph is randomly chosen in the relevant list. If the molecule is a free molecule, then all its fragments are generated and are finally put together to form a unique molecular graph.
- Then the random definition of the composition vector is realized. The attribution of the composition is done iteratively. At each iteration, a randomly chosen element gets a composition value. Picking randomly an integer value "i", the composition value will then be equal to $minValue + i * variation\ step$. Then the real range is updated and the next iteration is considered until a composition value is defined for each element of the mixture while satisfying a unit value for the sum of the composition. The appendix 10.5.1 describes more precisely this method.
- At last the operating conditions are set. Each one of them is randomly initialized to a value between the specified ranges.

This process is repeated until the initial population is complete.

4.6.4 Fragment creation

This section gives the details on how a free fragment is generated. This is a critical step as the performance of the genetic algorithm relies greatly on the diversity of the candidates. The molecular fragment graph generation is the responsibility of the GraphConstructor class. The sequencing of the different step is presented in Figure 66.

When a fragment is created, the first activity is the initialization **(1)**. It consists in the loading of all the user parameters. Then the GraphConstructor class is called to construct a random fragment graph that respects the user parameters.

The process of the graph construction begins with the initialization **(2)**: all the input data are treated. Then some random choices **(3)** are made to determine k, the number of functional groups in the fragment, and m, which determines the number of cycles in the fragment. The group vectors are generated with these two parameters and one group vector is chosen as a basis of the fragment graph construction following the method proposed by Korichi (2010). Using group vectors to build the graph is not necessary but it helps the algorithm to create more complex structures.

The elements are added sequentially one by one. If at one point the graph cannot be completed for a reason or another, the last element that has been added is modified. This process is detailed in appendix 10.4.

When an element is added, it can be a regular acyclic element or a whole cycle. The type of addition **(4)** is chosen randomly considering the number of cycles that remain to be constructed and the number and type of elements yet to be inserted.

If the addition of an acyclic element is chosen, then a functional group is randomly selected. After the insertion of the chosen group, the next element (acyclic element or whole cycle) is considered. This process goes on until the fragment is complete **(12)**.

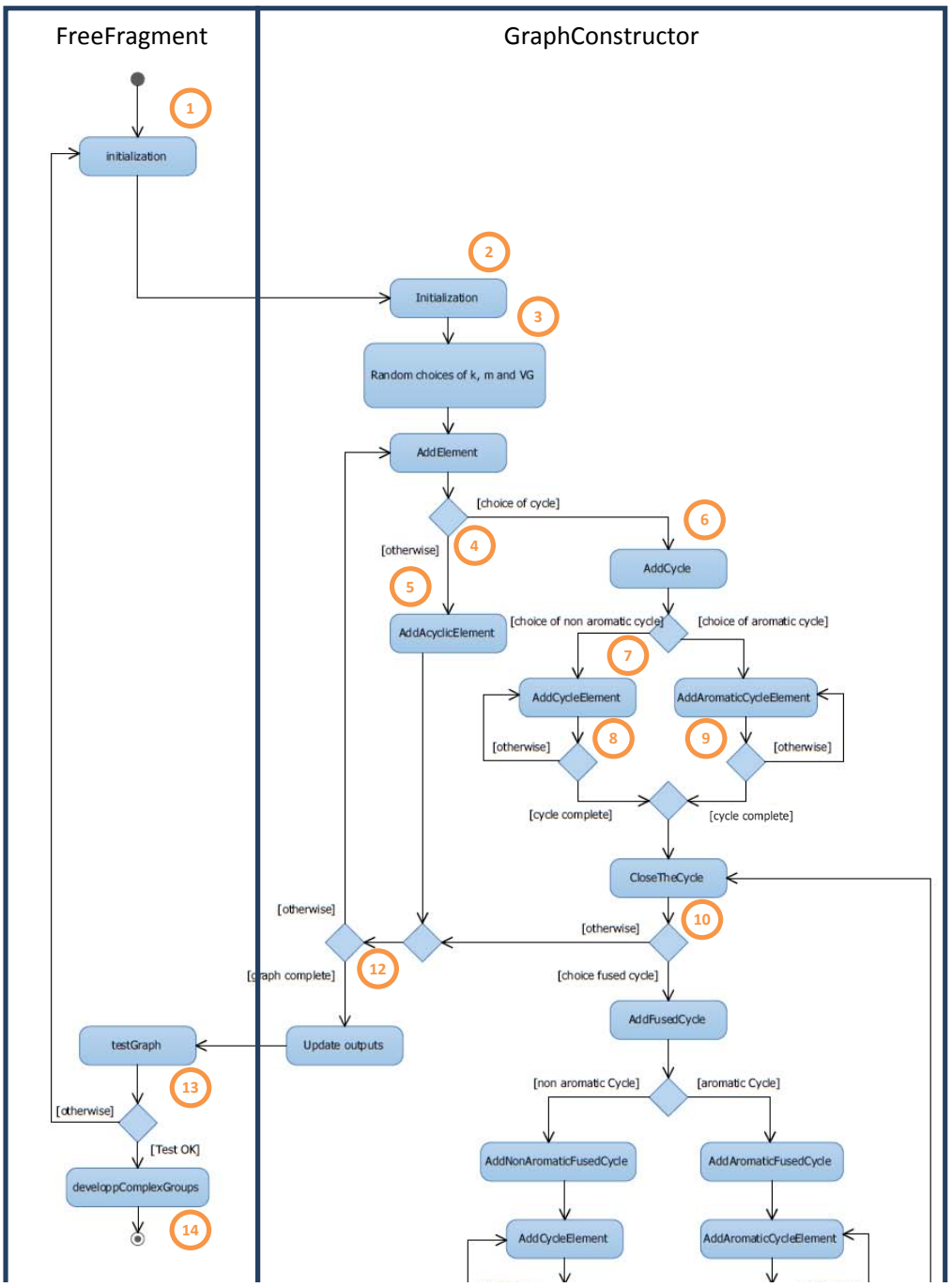


Figure 66: Activity diagram of the creation of a FreeFragment object

When a cycle is added **(6)**, all the elements that form the cycle are inserted one after the other. The branches that can be linked to the cycle are inserted only once the cycle is closed. This way the elements of a cycle are consecutive in the graph and thus it makes the graph easier to read. The size (number of

elements in the cycle) and the nature (aromatic or non-aromatic) **(7)** of the cycle are decided before the construction and the cycle elements are added one by one **(8) (9)**, in the respect of the nature of the cycle, until the cycle is complete. The graph is then closed **(10)**: this action corresponds to the connection of the last element of the cycle with the first element of the cycle. When the cycle is constructed, it is possible to add a fused cycle **(11)** on the new cycle. The decision is made randomly considering the number of cycles yet to be constructed and the remaining number of elements that can be inserted. Then it is randomly decided whether the fused cycle is aromatic or not.

For a non-aromatic fused cycle, the “attachment points” of the fused cycle are searched on the last inserted cycle and its adjacent cycles. A couple of “attachment points” is randomly chosen and the shortest way between these two points is determined. This path is then considered as a part of the fused cycle to build. Then the number of elements still to be added is randomly chosen and the elements are added one by one.

The process to construct an aromatic cycle is the same with more constraints. We have deliberately decided to limit ourselves to simple schemes to guaranty the aromaticity of the generated structures. More work needs to be done for being able to propose complex fused aromatic cycles. The additional constraints are the following:

- The “attachment points” must be consecutive in the last cycle.
- They must be connected with a double bound and must be potentially aromatic groups.
- There must be only one non carbon atom per aromatic cycle.

For both types of fused cycle, the groups are added one by one until the cycle is complete. The cycle is then closed **(10)** and the addition of another fused cycle is considered.

If it is chosen not to add a fused cycle, the next element is considered. This process goes on until the fragment is complete **(12)**.

When the graph is complete, the output data are updated following the formalism of the FreeFragment class. The graph and the other variables consistency are then tested **(13)**. If they are not, a new attempt of graph construction is made (NB: if the test failed several times in a row, then the search is stopped). When a graph passes the test, then all the complex groups it contains are expanded **(14)** in order to create an “expanded graph” only constituted of basic groups. This expanded graph is mandatory for the property evaluation procedure.

4.6.5 Mixture modification

Any genetic algorithm uses randomness, and probabilities are used to control the ratio of the different modification operators. In product design, many types of modifications are possible:

- a composition modification
- a molecule modification
- a fragment modification
- an operating condition modification

We have decided to extend the traditional use of probability to the management of mixture modification. With our method, probabilities are assigned by the user to rule the modifications of the mixture: composition, operating conditions, molecule itself.

The Figure 67 shows how a mixture is modified.

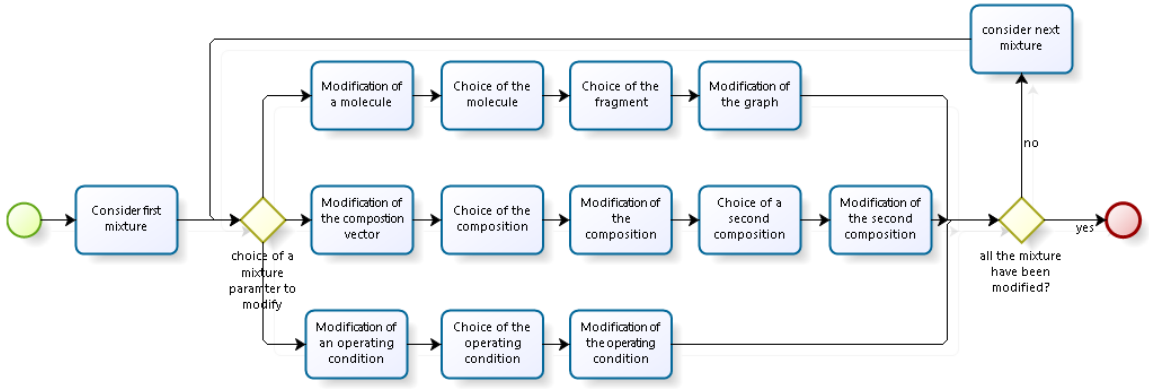


Figure 67: Simplified process of a mixture modification

As illustrated on Figure 67, if a molecule modification is chosen, then the algorithm chooses which molecule will be modified thank to the probabilities assigned by the user. A fixed molecule cannot be modified. A list molecule modification consists in choosing another graph in the associated list. A free molecule modification implies the modification of one of its fragments.

The modification of a fragment is similar to the modification of a molecule concerning fixed and list fragments but, for a free fragment, some modification operators are applied: mutation, crossover, insertion, deletion or substitution as proposed in chapter 3.

If the composition modification is selected, then the algorithm chooses which element will have its composition modified. It selects a new value for the composition that respects the user constraints, for

example the composition range constraint. Then in order for the sum of the composition vector to remain equal to 1, it selects one element (or more if necessary) and modifies its composition value so that the first modification is compensated (More details are given in appendix 10.5.2).

If the operating condition modification is chosen, then the algorithm selects which condition will be modified. It assigns another value respecting the user constraints.

4.7 DEPLOYMENT

The deployment is one of the last activities of a software development process. The purpose is to make the application available to the user. Three deployments are possible for our software application.

4.7.1 Restricted deployment

The restricted deployment only uses the search and the property calculation component (respectively the CAPD engineering service and the property calculation server). This configuration was used by the developers and by the different project partners when the MMI component (user interface service) was not available.

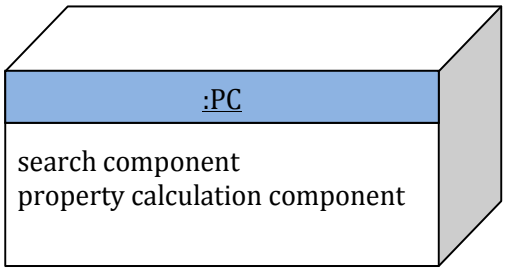


Figure 68: Deployment diagram of the restricted deployment

The entire execution is made on a single Windows personal computer.

4.7.2 Standalone deployment

The standalone deployment uses all the components of the final application.

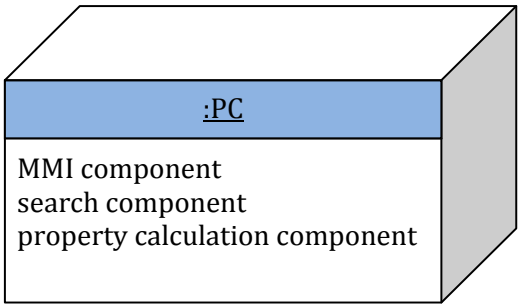


Figure 69: Deployment diagram of the standalone deployment

The entire execution is made on a single Windows personal computer.

4.7.3 Distributed deployment

With a distributed deployment, the application is distributed on different machines. Here the MMI component is on a personal computer and the search and calculation component are on a server.

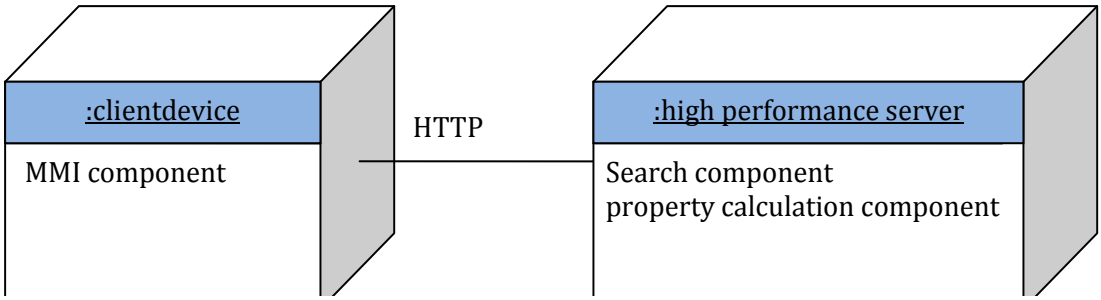


Figure 70: Deployment diagram of the distributed deployment

As the MMI is written in Java, any Operating System (OS) (Windows, Linux …) can execute it. The server hosting the two other components is a Windows calculation server that will have a greater computing power than a regular personal computer. This configuration thus enables the user to have the OS that he wants and a greater computing power.

4.8 VALIDATION

Some validation tests have been performed early in the software development with the first version of the software in order to validate our method in the first stage of development. As we have chosen to progressively add functionalities, this first prototype was already able to deal with free molecules that are built from a single free acyclic fragment. The objective function contains only calculable properties with no

operating conditions required. The application has been tested with very simple problems in order to compare the results to experimental data taken from the DIPPR data base.

Both of the following tests have been presented in (Heintz et al., 2011).

4.8.1 Test with a Tb=358.15K target

The mixture contains only one element and this molecule is built with one free fragment. The fragment is acyclic and it can be constructed with up to 10 functional groups among the following: CH3-, NH2-, OH-, Cl-, F-, Br-, I-, -CH2-, -CH-, >C<, CH2=, -CH=, >C=, CH≡, -C≡. The objective function is composed only of the boiling point temperature calculated with Marrero and Gani's first order group model (MG). The target value is 358.15K. The performance was calculated with a Gaussian function that was at the time expressed the following way:

$$G(P) = exp\left[-\alpha * \left(\frac{P - target}{2 * t}\right)^2\right] \quad (23)$$

We have chosen $t = 2$ and $\alpha = 0.001$. The algorithm consists of 30 iterations on one level, the population size is 100 and the elitism is 20. The genetic operator choice probabilities are 50% for cross-over, 20% for mutation, 10% for insertion and 20% for deletion.

The ten best molecules found are presented in the following table.

Table 6: Ten best acyclic molecules matching a 358.15K boiling point target

	molecule (SMILES)	T_b (MG)	$T_b$exp	Performance
1	C=CC#CC#C	358.08 K	NA	0.99999973
2	CC#CCC#C	358.24 K	353.7 K	0.99999944
3	FC#CCC#C	358.30 K	NA	0.99999854
4	NCCCF	358.84 K	NA	0.99997061
5	FC=CO	357.15 K	NA	0.99993720
6	CC=CO	357.09 K	NA	0.99993029
7	OCC#C	360.07 K	NA	0.99976953
8	ClC#CC#C	356.08 K	NA	0.99973128
9	ClCC=CC	360.57 K	356.7 K	0.99963449
10	ClCC#CC#C	355.51 K	NA	0.99914651

Considering that this first test is very simple, it is not surprising to find many candidate molecules with values close to the 358.15K target. Many molecules have no associated experimental data. But when the experimental value exists, the deviation is within the margin of error of the MG model.

4.8.2 Test with a Tb=358.15K and Tm=100K target

In this second test, the objective function is composed of two properties: the boiling point temperature with the same target value and the melting point with a target value of 100K. The other parameters are kept alike. Both properties have the same weight.

The ten best molecules for the second test are presented in the following table.

Table 7: Ten best molecules matching a 358.15K boiling point and 100K melting point target

	molecule (SMILES)	T_b (MG)	T_m (MG)	Performance
1	FC(Cl)C(C=C)=C	357.09 K	97.64 K	0.99979073
2	IC(F)Cl	368.26 K	98.05 K	0.99669527
3	ClC(F)C(Cl)=C	355.08 K	123.55 K	0.98266449
4	ClC(F)C(C=C)=CF	389.71 K	139.57 K	0.92321512
5	C=CI	346.16 K	167.64 K	0.87117724
6	BrCC=C	344.25 K	170.97 K	0.85896164
7	C=C(C=C)C=C	333.79 K	167.39 K	0.85826189
8	C=CCCl	319.16 K	166.73 K	0.83318863
9	BrC=C	307.55 K	158.86 K	0.82871597
10	BrCCC=C	375.75 K	182.17 K	0.81830087

The first two molecules match fairly well the constraints. Then, as expected from a more constrained problem than the first test, the candidate molecules have more widespread objective function values. It can be noticed that the best results are chlorofluorocarbons, which are molecules well known for having low boiling points and very low melting points.

4.8.3 Tests within the InBioSynSolv project

Due to confidentiality reasons, details are not provided. However one can say that 5 case studies have been considered within the InBioSynSolv project. Each time a single molecule is sought, aiming to match about 15 properties. To the date of this manuscript, some >40 property estimation models have been coded in the property calculation component, covering 24 different properties. The structure of the molecule itself is built by the search component from a mix of fixed, list and free fragments. The fixed or list fragments are sourced from seven pools of complex groups stored in the functionalGroupDatabase, coming from renewable resources. About 1.2 millions of chemicals have been generated and tested. Several original molecule candidates have been found thanks to the IBSS tool and are under further investigation by the Laboratoire de Chimie Agroindustrielle, by the Laboratoire de Chimie Organique et Macromoléculaire and by Rhodia).

4.8.4 Concluding remarks

These examples have shown that the research algorithm is able to give good results. Indeed, the molecular solutions proposed by the application are coherent with a good performance and highly diversified. Another validation test concerning a mixture is presented in chapter 7.

4.9 CONCLUSION

4.9.1 The IBSS tool development

The IBSS software solution implements the CAPD method proposed in the previous chapter, with all its functionalities.

The IBSS tool development relies upon four components: a search algorithm written in C#, a property calculation part written in VB.NET, a man-machine interface written in java and a functional group database stored as an XML file. Prior coding, UML2 have enabled to identify the users (basic and experts) and the main software functions listed afterwards. BPMN diagrams have described some dynamic behaviors between and within the software packages.

At the core component implementing the CAPD method, the search component has an object oriented and component oriented architecture. It uses a multi-level genetic algorithm and a molecular graph representation with a wide range of possibilities of specification.

4.9.2 The IBSS tool features

As a generic tool, it has a flexibility suited for many applications. Now comes a list of the main functionalities of the software, [X] indicating features unique to IBSS compared to other CAMD tools.

If we refer to the Problem package structure (Figure 48) with the algorithm, the mixture, and the level/objective function branches, we can list features:

For the objective function branch:

- Ability to choose target properties (min, max, bracket, set values)
- Ability to weight the properties in the objective function
- Ability to select all kind of models for the property estimation (similarity, QSAR, QSPR, group contribution of 1st order or higher order)
- [X] Ability to select the performance function (scaled, Gaussian, desirability function)

- [X] Ability to introduce the uncertainty of the property estimation method within the tolerance variable for the Gaussian performance function

For the mixture branch:

- [X] Ability to impose some of the mixture compounds (e.g. an active principle) and/or to seek some among a pre-defined list (e.g. a database of additives)
- [X] Ability to impose some of the mixture compounds composition (e.g. an active principle fraction) and the mixture operating conditions (e.g. temperature is set at 200K)
- [X] Ability to search at the same time the mixture compounds, their composition and operating conditions
- [X] Ability for each mixture compound molecule to fix or seek among a predefined list some fragments (e.g. a fragment sourced from renewable raw materials)
- [X] handling of lists of predefined fragments
- Handling of lists of predefined molecules

For the algorithm branch:

- [X] Ability to search for a mixture, incl. its compounds, its composition and its operating conditions (see above)
- Multi-level search with the ability to select different property estimation models for each levels

4.9.3 The IBSS tool perspectives

Among the functionalities identified in chapter 3, the penalization of unrealistic molecular structures is still under development. In parallel to the writing of this manuscript, a new functionality has been added. It concerns the use of experimental data for the evaluation of properties of a mixture. Indeed, we find necessary that our software could be able to use known properties of a molecule instead of calculate them with a property estimation model which may give an approximate value. This functionality remains to be tested.

To date, IBSS is deployed and is used by different user types. Three chemistry PhD students are using it as “basic user”. Two PostDoc in chemical engineering are using it as “expert user”. The Industrial partner will also be taught how to use it.

In order to give some numbers about the software application, the search component consists in 93 classes, 254 attributes and more than 28000 lines of source code.

THIRD PART

C. A DECISION MAKING PROCESS FOR CHEMICAL PRODUCT SUBSTITUTION

5.	State of the art	…….	109
6.	Proposition of a decision making process for chemical product design	…….	130
7.	Industry related case study	…….	175

ABSTRACT

With regard to the complex nature of product substitution issues in an industrial context, we wish to formalize a generic approach dedicated to address efficiently the situations where a product substitution is needed. We base our approach on concepts from different fields, i.e. model driven engineering, enterprise modeling, decision making processes and requirement management. These concepts are detailed first and the approach we propose is described just after. Finally, an industry related case study is presented. It illustrates how our decision making process and our CAPD tool can be used to find a greener solvent for printing facilities.

CHAPTER 5

State of the art

It rapidly occurred that a CAPD tool alone was not enough to guaranty that the best decision is made when it comes to the parameters setting or the choice of the final replacement product. This chapter has for purpose to define the main concepts needed for the formalization of a decision making process dedicated to the chemical product substitution in an industrial context. We rely on concepts from enterprise modeling, model driven engineering and decision making fields.

In the frame of our work on Computer Aided Product Design within the InBioSynSolv ANR project, we have noticed that all our project partners contributed during the phase aiming at setting the parameters of the problem. Indeed, product specifications are mostly given by the industrial partner, while our chemical engineer team uses internal competencies to set concrete and calculable property targets and the chemist partners define the constraints on the chemical structures of the molecules. Nevertheless, difficulties arose: time was needed for each partner to explain to the others its own understanding of the problem. This is further complicated by the fact that each expert has its own semantic language.

The collaborative meetings are relatively easy to perform with a few partners as in our case. However, formalization is useful as it can improve the collaborative process efficiency in small scale projects and it is necessary for large scale problems like the ones occurring in a chemical international company.

The chemical companies are nowadays in a strongly competitive context where regulations like REACH or the VOC directives are pushing them to investigate new molecules. The CAPD tool presented in the previous chapters allows a more effective, quicker and less expensive investigation than other methods to find new molecules. But in order for our tool to provide solutions compliant with the enterprise needs and objectives, it is necessary to involve all the actors taking part in the chemical product substitution process, who are coming from different levels of the organization. We propose to formalize a decision making process for the substitution of products in a chemical related enterprise. This process integrates our tool and relies on the different concepts coming from system and enterprise engineering and decision theory.

We can superimpose our decision process with the chemical supply chain as defined by (Marquardt et al., 2000) and sketched in Figure 71.

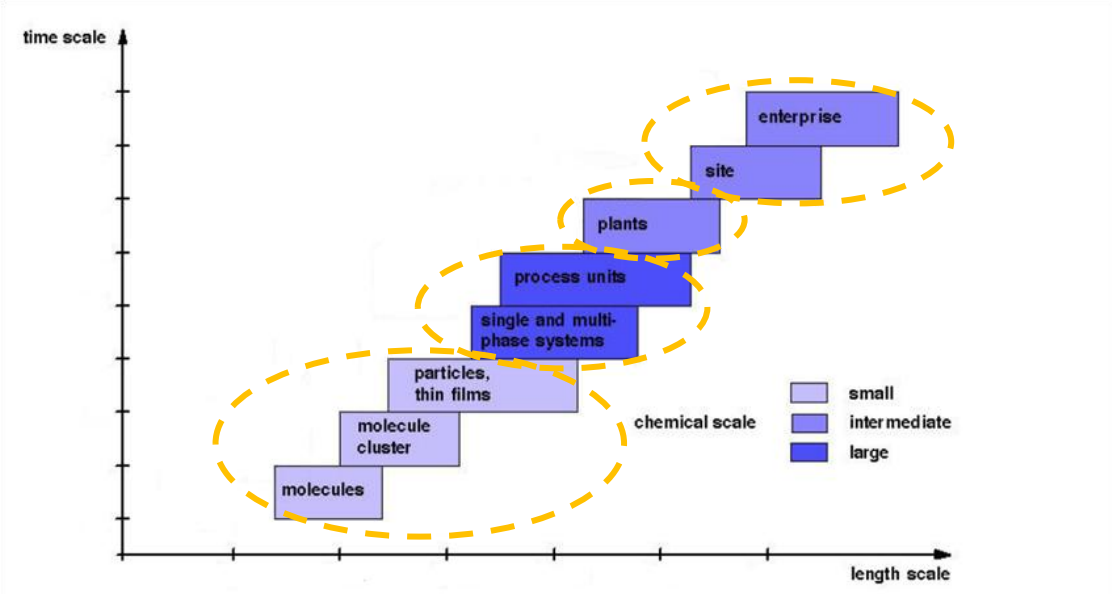


Figure 71: Chemical supply chain inspired by (Marquardt et al., 2000)

Our decision making process encompasses all the scales of the supply chain aggregated in four blocks: one at the small chemical scale, one at the process unit scale, one at the plant unit scale, and finally one at the production site and enterprise scale. By considering the entire chemical company, this decision process goes a step further in the use of the CAPD tool described in chapter 3 and 4. Indeed, it goes beyond the molecular aspects and considers the entire chemical supply chain. We have highlighted with dashed circles in Figure 71 the four main blocks that we are going to reuse in the next chapter.

5.1 ENTERPRISE MODELING

Our framework must be integrated into the environment of the chemical enterprise. In this section, we define formally the enterprise and introduce the fundamental of enterprise modeling in order to use its main concepts in our framework.

5.1.1 Definition

According to Izza (2006), an enterprise is the place where diverse activities are performed. Each activity aims at creating value, and information is considered as a vital resource for its functioning. The enterprise activities are performed by actors or by resources. To reach the aims and the objectives of the enterprise, their orchestration is essential.

According to the report of the PROLOG specification (AS n°35 PROLOG) of the GDR MACS (LAAS-CNRS, 2003), these activities can be done in a similar way in a project, in a traditional enterprise

or in a network of enterprises. The achievement of these collective activities in a specific domain is the very definition of the business of an enterprise according to Martin et al.(2004).

Through these different definitions, it can be seen that an enterprise is a complex system of systems. In order to analyze the properties of an enterprise, the characteristics of these sub systems must be defined.

5.1.2 Enterprise modeling

Enterprise modeling aims at building a model including all or a part of the enterprise. The enterprise is then seen as a system and its modeling must explain its structure, its organization and its functioning (Pourcel and Gourc, 2005). The model must give a representation of the enterprise architecture in order to ease its comprehension. Enterprise modeling is also seen as the art of externalization of the “know-how” of the enterprise (Touzi et al., 2009), making it a prerequisite for all enterprise integration approaches.

For representing the complexity of an enterprise in an understandable manner, multiple points of views are necessary. According to the IEEE standard, Recommended Practice for Architectural Description of Software-Intensive Systems (IEEE, 2000), a view is a representation of the integrality of a system through a specific perspective of a set of linked interests. As described in the norm ISO TC 184/SC 5 (2000), four views can be considered:

- The functional view describes the processes and their structures. The names, goals and actions of the processes can be identified.
- The informational view forms the data-flow. This view indicates which documents and data are used at each stage of a process. It describes the system objects, their relationships and their different possible states.
- The organizational view defines who is responsible or capable to carry out the process.
- Lastly, the resource view or operational view specifies the tools or systems allowing the fulfillments of the process by describing the human resources and necessary equipment as well as the type of resource management.

Each of these views manipulates its own concepts and can be expressed through different formalisms. However the global consistency must be kept in mind and guarantied. The views do not have the same importance during an enterprise modeling approach.

The Figure 72 from Touzi (2008) represents the four views and highlights the formalism presented by Vernadat (1999).

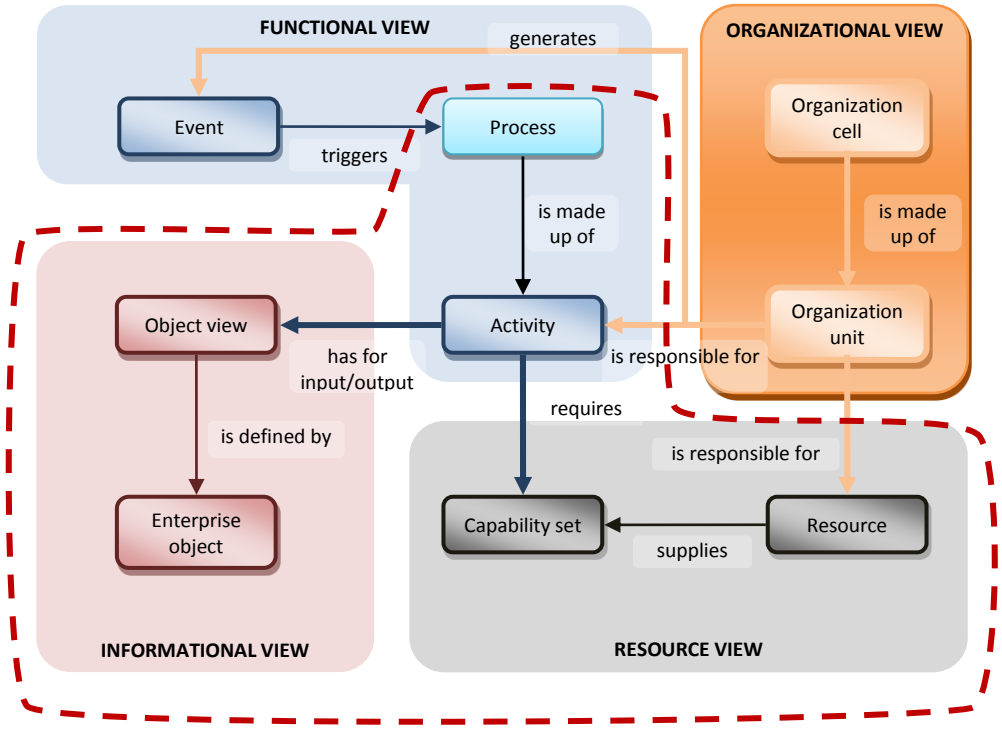


Figure 72: The four views of the enterprise from (Touzi, 2008)

For our approach, we focus on the dashed red contour outlined in Figure 72, encompassing the informational view, the resource view and a part of the functional view.

5.2 MODEL DRIVEN ENGINEERING

In order to further formalize our frame and in accordance with Model Driven Engineering, we propose to associate a model and a metamodel of the constraints defined by each person involved in the setting of the parameters. Here, we define Model Driven Engineering. More information can be found in Ulmer (2011)

5.2.1 Definition

Model-Driven Engineering (MDE) is a “general integrative approach” (Favre et al., 2006) (Perez et al., 2008) (Combemale, 2009) which makes tools, concepts and languages available for creating and transforming models. It is an evolution of Model-Driven Architecture and an initiative of the Object Management Group (OMG). It allows the integration of different technical spaces which can be object oriented technologies (using UML) or structured documents (using XML).

The majority of the modeling methods define contemplative models, essentially used in an aim of communication and understanding between human agents/actors. These models need to be transformed to be understood and compiled by a computer. To achieve that, it is necessary to formalize the models, the transformations to which they are subjected, the representation languages and the metamodel to which they are associated. The main idea proposed by MDE is to be able to use as many Domain-Specific Modeling languages (DSML) as required by the technological aspects that are used. The MDE proposes also a systematical use of metamodels, models and processes of design which are precise and formal enough to be interpreted and transformed by computers.

5.2.2 Real system, model, metamodel

The links between the real system, model and metamodel are represented in Figure 73. The level M0 contains the real system. At the M1 level, a model represents a simplification of this previous system. This model must be compliant with an expression language which is defined by its metamodel at the M2 level (Bézivin, 2004)(Favre et al., 2006).

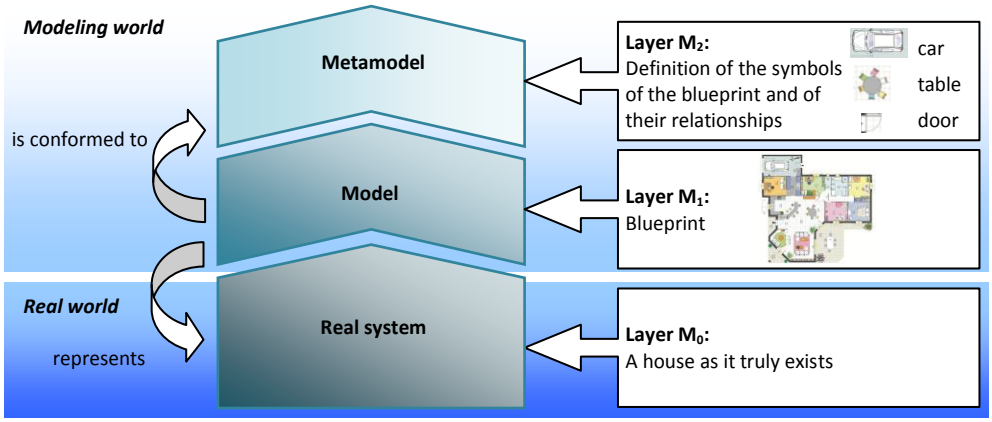


Figure 73: the real system, the model and the metamodel

Figure 73 shows a simplified representation of a house through the different abstraction layers. In this example, the real system is the house as it truly exists (M0 layer). It is represented by an architectural drawing (M1 layer). This drawing uses a legend (symbols like graphical elements to sketch a table, a door, a car, etc.) and conventions of representation (constraints like a scale, colors per room usage, etc.) which are defined by the metamodel (M2 layer). For information, the abstraction goes until the metametamodel (M3 layer) which defines the language in which the metamodel is expressed.

5.3 DECISION MAKING PROCESS

The goal of our frame is to provide a structure where the decisions about the substitution of molecules can be made in the best and most efficient conditions. We present in this section how we define a decision and what are the most used decision making processes.

5.3.1 Definition

Mintzberg et al. (1976) define a decision as a specific commitment to action, which usually means a commitment of resources.

Simon (1960) introduces the notions of programmed and nonprogrammed decisions:

- A programmed decision is a repetitive decision where the process is clearly defined.
- A nonprogrammed decision is a novel decision where the process is ambiguous.

These are not distinct types but they define a continuum where every type of decision can be placed. In a more recent literature, the vocabulary has evolved, becoming structured and unstructured decisions (Mintzberg et al., 1976; Aurum and Wohlin, 2003; Cauvin, 2005). Cauvin (2005) goes further by specifying the decision system, the design situation and the production situation for each type of decision. Their relations are reported in Table 8 and provide various situations, from routine to creative for the design and for the production.

Table 8: Types of decisions and associated situations (from Cauvin, 2005)

Type of decisions	Slightly structured	Moderately structured	Highly structured
Type of the system of decision	Human		Artificial
Design situations	**Creative design** situations (new products)	**Innovative design** situations (known products)	**Routine design** situations (variants)
Production situations	**highly disrupted** situations	**moderately disrupted** situations	**not disrupted or slightly disrupted** situations

Basically, a decision can be represented by different types of format:

- The decision table
- The graphical diagram (decision tree, inference diagram)

- The declarative (natural or semi-natural) language
- The specific (standard or proprietary) modeling language

Our main objective is to structure the currently unstructured decision that is the decision of the substitution of a problematic molecule. For this purpose we have to formalize a decision making process.

5.3.2 Decision making process model

5.3.2.1 Simon's IDC model and derivatives

A decision process is a set of actions and of dynamic factors that begins with the identification of a stimulus for action and ends with a specific commitment to action (Mintzberg et al., 1976).

Simon's IDC (Intelligence, Design and Choice) (Simon, 1960) is the most common decision process. It consists in three phases:

- The Intelligence phase: identification and understanding of the problem
- The Design phase: design and identification of alternative solutions
- The Choice phase: selection of one of the alternatives

Vallin and Vanderpooten (2000) have added a fourth phase: the Implementation phase which deals with the operational Implementation. Eventually corrections and validation are needed and then the process starts over. The process is schematized in Figure 74.

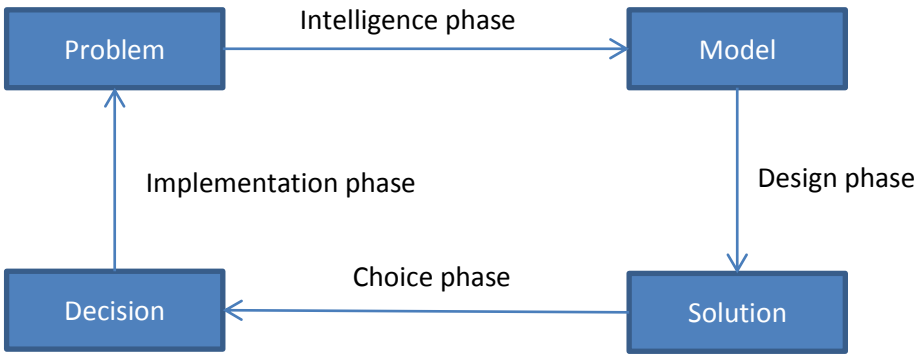


Figure 74: Outline of the decision making process (from Vallin and Vanderpooten, 2000)

When a problem occurs, the Intelligence phase identifies the main parameters and models it. The model is then used to produce alternatives or solutions during the Design phase. The Choice phase

consists in the analysis of the solutions and the choice of a specific one which becomes the decision. Finally, the actions associated to the decision are performed during the Implementation phase.

Following the expanded model of Huber (1980), Ashrafi (1998) proposes an additional phase: the Monitoring phase. This phase is the feedback and control phase. It comprises maintenance, support and updates. Cauvin (2005) introduces the knowledge capitalization as a final phase as shown Figure 75. This allows reusing the analyses made for a decision in future decision making processes.

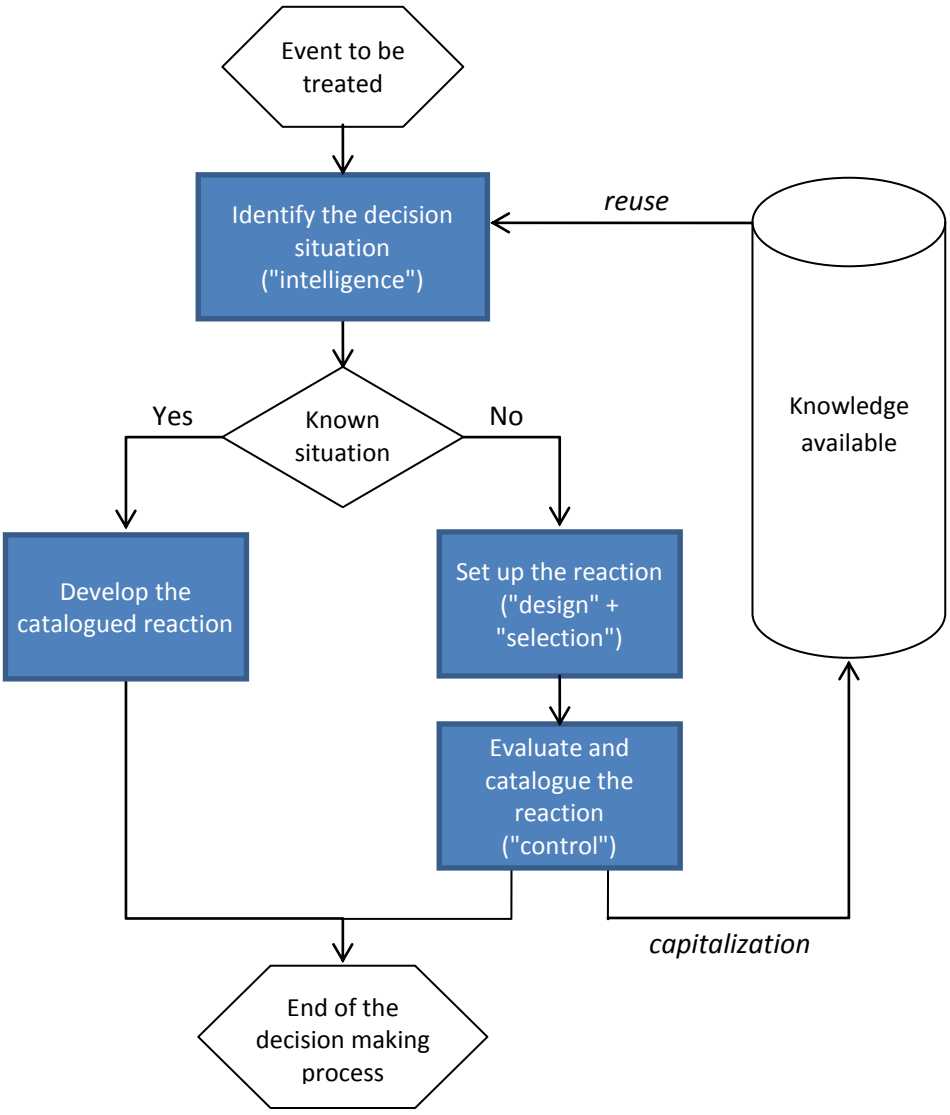


Figure 75: Cauvin's decision process (from Cauvin, 2005)

5.3.2.2 Howard's decision analysis procedure

In Howard (1966), the definition of a decision is restricted to "an irrevocable allocation of resources". Compared to the IDC model (Simon, 1960), the decision analysis procedure is more oriented toward business decisions dedicated to profit making. It has for specificity that the alternatives cannot be automatically generated and that uncertainty must be taken into account. The decision analysis procedure of Howard (1966) can be simplified the following way:

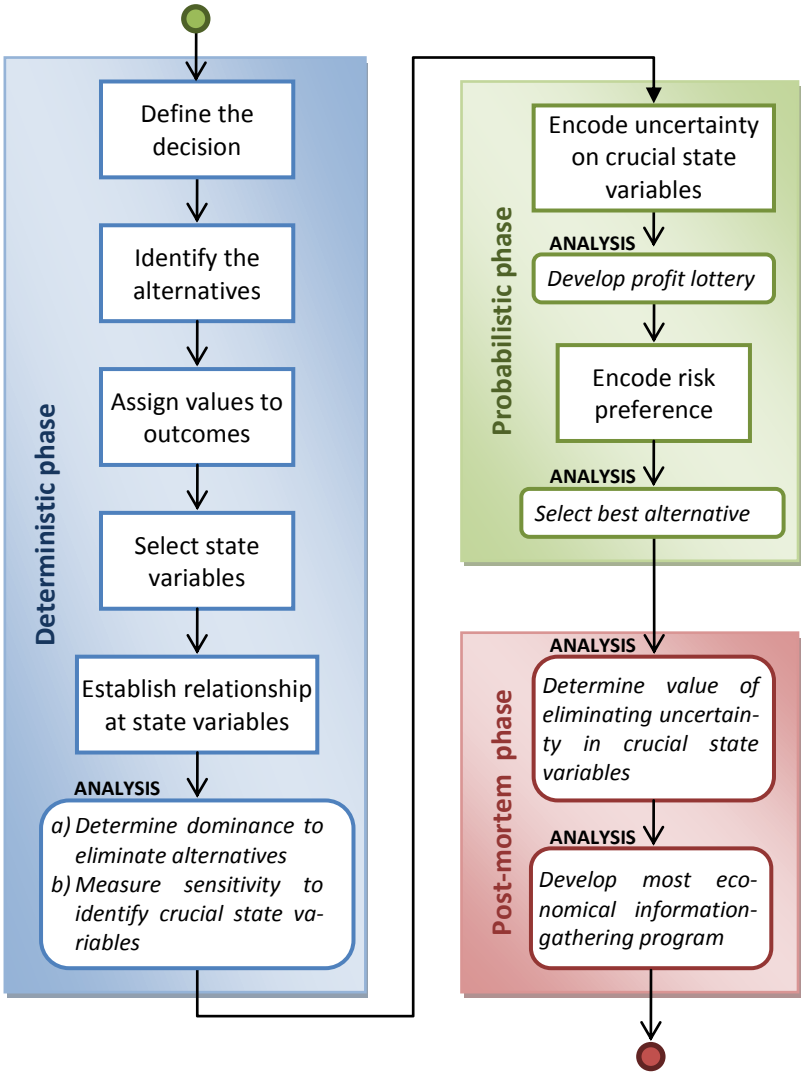


Figure 76: Howard's decision analysis procedure (Howard, 1966)

The first step of the deterministic phase consists in defining which decision is to be made. Then, the possible alternatives are identified and the criteria of good or bad outcomes are defined. With these criteria, the state variables are determined. These are the variables of the problem which will impact the values of the outcomes. Once the state variables are determined, nominal values and ranges of variation are assigned to them. Finally the relationships between the different variables (dependencies) and

between the variables and the value of the outcomes are established for each alternative. At this step, a deterministic model of the decision problem is available. It is possible to perform analyses with this model. The first one consists in sweeping the state variables through their range of values in order to select promising alternatives. The second one aims to identify the crucial state variables through a sensitivity analysis.

Following the deterministic phase, the probabilistic phase begins. Using the current state of knowledge, probabilistic distributions are assigned to the state variables (taking into account the relationships between the variables). They represent the uncertainty on the variables. Once this is done, an analysis can be performed to determine the probabilistic distribution of the profit associated to each promising alternative. This is named “profit lottery” of the alternatives. According to a risk preference, a best alternative is chosen. But the decision making process does not necessarily stop there.

During the post mortem phase, the impact of the uncertainties of the variables on the profit of the selected alternative is evaluated. It allows finding the variables for which it would be profitable to eliminate or at least reduce uncertainty by gathering more information. With this new knowledge, the probabilistic distributions are modified and the currently selected alternative might no longer be satisfactory. If it is the case, the probabilistic phase starts again with the new probabilistic distributions.

5.3.2.3 Multi Criteria Decision Analysis/Making

Multi Criteria Decision Analysis (MCDA) or Multi Criteria Decision Making (MCDM) has for objective to support the decision makers facing a problem with antagonistic criteria. For this type of problems no optimal solution exists. One of the methods is to consider the set of nondominated solutions. A solution is nondominated when, in the entire solution space, there is no solution that is better for every criterion. This set of solution defines the Pareto optimal solutions. The decision makers have then to make a trade-off by defining their preferences. (Baez Senties et al., 2010) uses a genetic algorithm to define the Pareto front.

Other methods are multi-level approaches. The main difference is that the preferences are made a priori. Homburg (1998) has established a hierarchical procedure where the general preferences are expressed on a top-level and the compromises are made on a base-level in order to find a final compromise solution. Thery and Zarate (2009) propose a decision making structure dedicated to energy planning. Decision making levels are defined according to a space/time scale and a top-down approach where the decisions made at a specific level become constraint for a lower level.

5.3.3 Enterprise decision making

In the context of enterprises, different kinds of decision can be identified. There are three categories of decisions according to Ansoff Model (Ansoff, 1965): the strategic, the administrative and the operational decisions. They are represented in Table 9. His approach is very economic oriented. Nowadays sustainability issues should be integrated to this model by considering environmental and societal aspects.

Table 9: Ansoff's decision making model (from Ansoff, 1965)

	Strategic	Administrative	Operating
Problem	To select product-market mix which optimizes firm's ROI potential	To structure firm's resources for optimum performance	To optimize realization of ROI potential
Nature of problem	Allocation of total resources among product-market opportunities	Organization, acquisition and development of resources	Budgeting of resources among principal functional areas Scheduling resource application and conversion Monitoring and control
Key decisions	Objectives and goals Diversification strategy Expansion strategy Administrative strategy Finance strategy Growth method Timing of growth	Organization: structure of information, authority, and responsibility flows Structure of resource conversion: work flows, distribution system, facilities location Resource acquisition and development: financing, facilities and equipment, personnel, raw materials	Operating objectives and goals Pricing and output levels Operating levels: production schedules, inventory levels, warehousing, etc. Marketing policies and strategy R&D policies and strategy Control
Key characteristics	Decisions centralized Partial ignorance Decisions nonrepetitive Decisions not self-regenerative	Conflict between strategy and operations Conflict between individual and institutional objectives Strong coupling between economic and social variables Decisions triggered by strategic and/or operating problems	Decentralized decisions Risk and uncertainty Repetitive decisions Large volume of decisions Suboptimization forced by complexity Decisions self-regenerative

In this table, three types of decisions are introduced:

- The strategic decisions: they are the most important as they determine the general orientation of the enterprise.
- The administrative decisions: they are the prolongation of the strategic decisions and command the operational directions

- The operational decisions: they are the decision of the common management and correspond to the least important decisions.

Anthony (1965) has a similar approach when he proposes a framework for planning and control systems. This framework must facilitate the decision making process. It is made of three main activities: the strategic planning, the management (tactical) control and the operational control. The strategic planning is defined as the process deciding the objectives and the policy of the organization. The management control is the process by which the managers ensure that the resources accomplish the organization objectives. The operational control is defined as the process of assuring that specific tasks are carried out. These three activities are supported by the information handling and the financial accounting. Relationships between the activities exist and they are indicated on Figure 77.

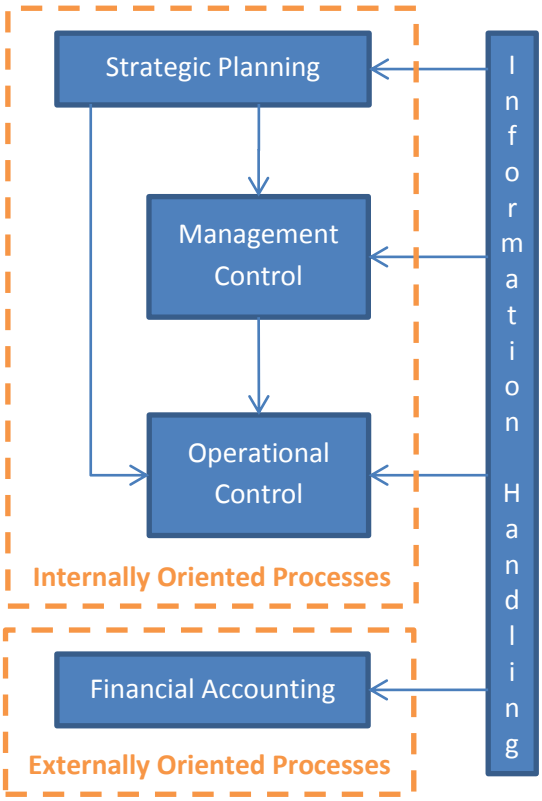


Figure 77: Planning and control processes in organization (Anthony, 1965)

The management control is influenced by the strategic planning activity as its goal is to reach the strategic objectives. It also influences the operational control as it imposes the tasks to be performed.

These two approaches by Ansoff and Anthony are very similar and highlight the propagation of the decision making into the organization of an enterprise. Our opinion is that a formal enterprise modeling will facilitate this propagation and is necessary for our frame.

5.3.4 Decision making methods

Once all the alternatives are generated, the choice of the alternative that will be implemented must be made. For multi-criteria problems or group decision making, choosing the best alternative can be difficult. To help the decision makers, many decision making methods exist, as for example multi-attribute utility technique (MAUT) (Keeney and Raiffa, 1993), ELECTRE (Roy, 1968), DELPHI (Dalkey and Helmer, 1963), PROMETHEE (Briggs et al., 1990), Analytic Network Process (ANP) (Saaty, 2001) and Analytic Hierachy Process (Saaty and Vargas, 1994). We have chosen to focus on the DELPHI method.

Linstone and Turoff (1975) define DELPHI in a very general manner as “a method for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem”. More concretely, this method consists in soliciting experts and making them answer questionnaires individually. A facilitator then gives an analysis on the questionnaires answers and on the motivations of these answers. Then a second round starts and the experts answer the questionnaires once again but this time, in light of the analysis of the previous round, their answers may be different. The rounds go on until a stop criterion is satisfied (consensus or number of rounds). If no consensus is reached, the median scores of the final round determine the results. This method relies on the hypothesis that this process of communication of the results will lead the group to converge toward the correct answer.

Many variants of the DELPHI method exist such as DELPHI-SWOT in (Tavana et al., 2012). Some are reformulations and others are mixed method combining the DELPHI method with other decision making methods.

5.4 REQUIREMENTS MANAGEMENT

The Intelligence phase of the decision process for the substitution of a molecule consists mainly in the determination of the requirements on the replacement product. In this section, the requirement and business rule concepts are defined and the requirement engineering process is presented.

5.4.1 Definition

Requirements Engineering originally comes from Software Engineering but has been generalized to system engineering. According to Wiegers (2009), there is a lack of general agreement in research as to what is a requirement. Indeed several definitions exist. The IEEE Standard Glossary of Software Engineering Terminology (IEEE, 1990) defines a requirement as "(1) A condition or capability needed by a user to solve a problem or achieve an objective, (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents, or (3) A documented representation of a condition or capability as in 1 or 2". Coulin (2008) interprets this formal definition by identifying a requirement as something that the system must do, must have or must satisfy as determined by someone related to its development. Wiegers (2009) goes further by encompassing both the users' view of the requirements and the developer's view.

Hellouin (2002) gives a less software development centered definition. "A requirement […] merges the needs, the requests, the wishes and the constraints. These words can be employed to express a requirement according to the entity who expresses it, wishes it, or undergoes it. A requirement defines an expectation expressed by one of the stakeholders in a direct (performance) or indirect (forbidding, limitation) manner." We choose this definition and add that a requirement is a formal and technical expression of a need or of a constraint (Konaté, 2010).

5.4.2 Business rules

Business rules are a special type of requirements. Indeed requirements are usually constraints that guide the design of a system. Once the system is created, modifying one of the requirements is difficult and sometimes impossible. Business rules define the control part of the requirements and can be modified at any moment. They rely on the definitions of terms and facts. Business rules can be used to represent both user requirements and conditions to which the system should conform (Wan-Kadir and Loucopoulos, 2004).

The main concepts of business rules appeared in the 80's and are today defined as "a statement that defines or constrains some aspect of the business" (BRG, 2000). Ross (2003) identifies three categories of rules:

- Rejectors: if an event causes the violation of this rule, then the event is not allowed.
- Producer: this rule produces value.

- Projector: this rule causes new events.

The OMG (Object Management Group) specification makes a difference between the structural rules which claim a necessity and the operative rules which claim an obligation. The rules can thus have different level of enforcement.

Business rules are volatile concepts (Wan-Kadir and Loucopoulos, 2004). They allow externalizing control from the processes or procedures (Ross, 2003). The rules can be established in a separate rule layer or component which enables a direct management of the rules. Software tools have been developed in order to manage this layer within any enterprise Information System. Those tools are generally called Business Rules Management System (BRMS). Among them, the most famous are IBM-ILOG JRules, OpenRules, OpenLexicon, or Drools. Those solutions integrate a specific component, the Business Rule Engine (BRE), which relies on rules repository. Figure 78 is IBM's overview of its BRMS ILOG JRules.

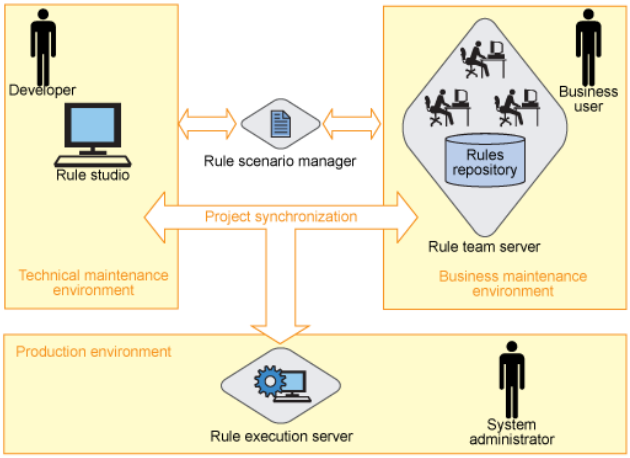


Figure 78: ILOG JRules

In this specific BRMS, the business rules are managed and stored in the business maintenance environment. The technical maintenance environment is responsible for support and debugging whereas the production environment is the BRE as it executes the rules. The rule scenario manager is here to simulate business policy changes.

5.4.3 Requirements engineering process

According to Nuseibeh and Easterbrook (2000), a requirements engineering process consists of five core activities:

- Eliciting requirements: gathering and interpretation of information about the requirements.
- Modeling and analyzing requirements: construction of an abstract description.
- Communicating requirements: documentation of the requirements for a better communication between the stakeholders.
- Agreeing requirements: validation of the requirements by all stakeholders in particular by those who have divergent goals.
- Evolving requirements: management of the requirements changes

Our proposal focuses on those activities except the “evolving requirement” one.

A considerable part of requirements engineering relies on the social and psychological aspects for improving the requirement process. For example, they allow answering questions such as how to motivate the people involved in the process or how to avoid withholding of information. These aspects are out of the scope of our work. We simplify the problem by considering perfect stakeholders (competent, motivated and meticulous).

Aurum and Wohlin (2003) consider requirement engineering as a decision making process. Indeed, the stakeholders have to decide of a set of requirements. They combine Macaulay’s Requirement Engineering model with two decision-making models of Anthony (1965) and Mintzberg et al. (1976).

5.5 MODELING LANGUAGES

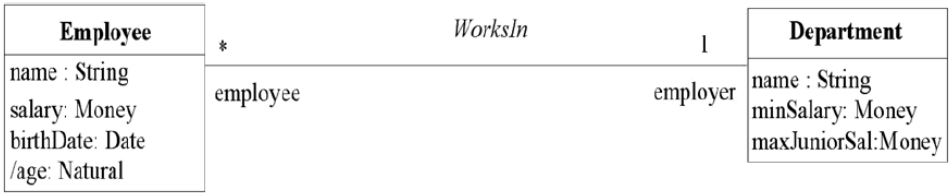
Coming from Object Oriented world and semantic, web and interoperability fields, many modeling languages/notations/techniques that can be valid for requirements management exist such as UML2, SysML, OCL, SBVR, OWL, RIF, RuleML, SWRL, BRML, PRR, SRML, ISO CL…

Those modeling languages allow expressing requirements without ambiguity, which is not the case with natural language. The drawback is that it generally requires some mathematical background. We chose to focus more particularly on OCL and SBVR. Those two languages are presented hereafter.

5.5.1 Object Constraint Language

The Object Constraint Language (OCL) (OMG, 2006) is “a formal language [that remains easy to read and write,] used to describe expressions on UML models”, particularly on class diagrams. It has been developed at IBM and is now part of the Object Management Group standard. The structural constraints are expressed thanks to invariant conditions and the constraints on behavioral features are

specified thanks to pre- and postconditions. These expressions cannot directly modify the system or trigger any action, contrary to what business rules can do. Here follows an example taken from Cabot and Teniente (2007).



- **context** Department **inv**:

  self.employee->forAll(e|e.salary>self.minSalary)

- **context** Employee **inv**:

  self.salary->self.employer.minSalary

- **context** Department **inv**:

  self.employee->select(e|e.salary<=self.minSalary)->size()=0

The three OCL expressions represent the same constraint: the salary of an employee must be higher than the minimum salary of his/her department.

5.5.2 *Semantics of Business Vocabulary and Rules*

The SBVR (*Semantics of Business Vocabulary and Rules)* (OMG, 2008) is a fairly new standard for specifying business objects and rules. SBVR describes structural aspects of business processes, as well as the policies that should guide agents' behavior in certain situations (Solomakhin, 2011). The structural business rules are expressed using alethic modal operators ("It is possible that …") and the operative business rules use deontic modal operators ("It is permitted that …"). The SBVR metamodel is based on two main features:

- the business vocabulary which defines
  - noun concepts: "concept that is the meaning of a noun or noun phrase"
  - fact types or (verb concepts): "a concept that is the meaning of a verb phrase" and represents the relationships between noun concepts
- the business rules which can be structural rule or operative behavioral rule as already defined in section 5.4.2

(OMG, 2008) uses SBVR Structured English to define the SVBR vocabularies and rules but adds that it is just one of the possibilities. SVBR Structured English is presented in Annex C of (OMG, 2008). It defines key words, logical operations and modal operations. In this Annex, the following example can be found, which applies to a fictitious car rental company with branches in several countries. This example includes three key words or phrases ('keyword' font, i.e. red text), two designations for noun concepts ('term' font, i.e. dark cyan and underlined text) and one for a fact type ('verb' font, i.e. blue italic text):

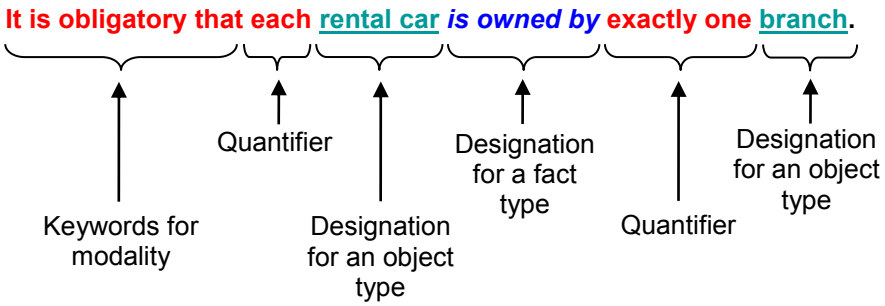


Figure 79: SVBR Structured English example

Another famous business rules language is BRS RuleSpeak (Ross, 2003) which is based on rule sentence templates (must, may, need, can…). This language is older than SBVR as it started in 1996. Contrarily to SBVR, it is only a business rule notation and does not provide a frame for expressing business vocabulary. RuleSpeak can however replace SBVR Structured English in SBVR rule definition. The RuleSpeak version of the example given for SBVR Structured English is the following:

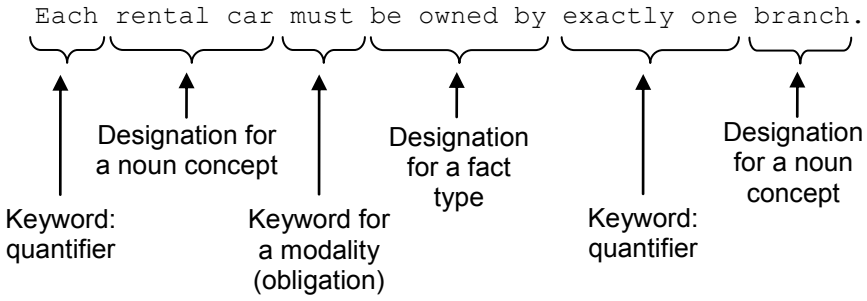


Figure 80: RuleSpeak example (from www.brcommunity.com)

It can be noticed that RuleSpeak and SBVR using Structured English are very similar.

5.6 CONCLUSION

In this chapter, we have introduced some concepts that we are going to use for building a formalized decision making process for the design of product in a chemical related enterprise.

Firstly, we have presented how an enterprise can be defined, and how it can be modeled. In particular, we have seen that four views enable to represent an enterprise and its complexity: the functional view, the informational view, the organizational view and the resource view.

Second, we have introduced the concepts of model driven engineering, which we will use in order to propose a formalized frame. We have detailed, in particular, the notions of real system, model and metamodel.

We have then presented how to define a decision, what are the most common decision making processes, and what decision making methods exist. We will keep in mind the three phases, “Intelligence”, “Design” and “Choice”, in the decision making process proposed by Simon (1960), to which a fourth phase “Implementation” has been added by Vallin and Vanderpooten (2000). We have also introduced the decision analysis procedure of Howard (1966), and in particular the “Post-mortem” phase, which consists in reducing the uncertainty by gathering information. Concerning the decision making methods, we have detailed on the DELPHI method.

Within the Intelligence phase, we have focused on concepts associated to requirements management. In particular, we have detailed more thoroughly the business rules. We have also introduced the requirement engineering process activities.

Finally, we have presented some modeling languages which can be used for requirement management. In particular, we have detailed the Object Constraint Language (OCL) which has for purpose to express constraints on UML models, and particularly on class diagram. We have also focused on the Semantic for Business Vocabulary and Rules (SBVR), which can be used for expressing business rules.

The next chapter uses these concepts and languages to propose a formalized decision making process for the design of products in the context of a chemical related enterprise.

CHAPTER 6

Proposition of a decision making process for chemical product design

This chapter presents our proposal for a decision making process based on the concepts introduced in chapter 5. We build a three-phase process by adapting Simon's approach to our problem. In the Intelligence phase, we guarantee an alignment of the requirements of business players issued from different levels of the enterprise and having different fields of expertise. In the Design phase, alternatives satisfying these requirements are generated using a product design method, possibly our CAPD tool. The Choice phase is divided in two stages, one for determining which promising alternative is going to be consolidated by laboratory testing, and one for determining if this alternative will finally be chosen in light of the laboratory results.

6.1 DECISION MAKING PROCESS OVERVIEW

Our decision making process model is inspired by the decision process proposed by Vallin (Vallin and Vanderpooten, 2000). The last step, the Implementation phase, is out of the scope of this PhD. We thus focus only on the first three phases, which reduces the model to the decision process model of Simon (Simon, 1960). The process is summarized on Figure 81.

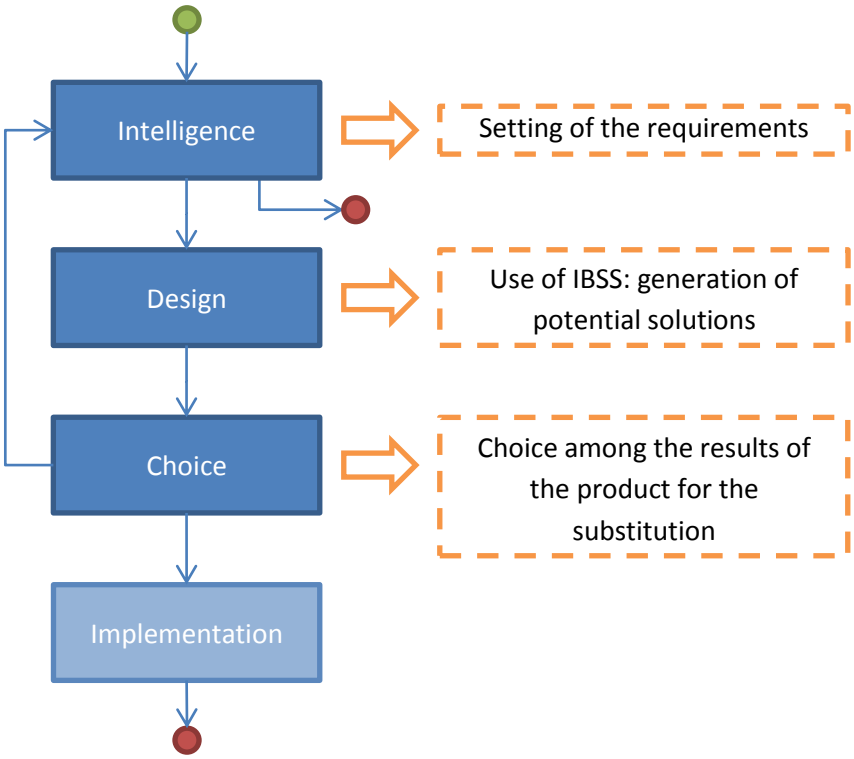


Figure 81: Overview of our decision making process

Let us consider a general chemical product substitution problem. A stimulus, for example a regulation evolution, imposes the enterprise to adapt itself. This leads the project director to launch our decision making process. During the Intelligence phase, the stakeholders decide if the substitution of a chemical product is necessary. If not, other actions than the substitution must be undertaken, and this particular decision making process stops. Otherwise, they define the requirements for the future chemical product. These requirements hence constitute a model of what is wanted. The Design phase consists in the generation of candidate solutions, called alternatives, by using the requirements resulting from the Intelligence phase. During this phase, any method can be used, but in this PhD only the use of our CAPD tool is presented. In the Choice phase, the DELPHI method is used in order to select the best option. If the experts involved in the DELPHI method judge that the alternatives proposed are unsatisfactory, then

the people of the Intelligence phase are asked to reconsider their requirements. Finally when an option is selected, the Implementation phase begins.

The main challenges are to define a structuring frame for each phase and to propose a formal approach of decision analysis in order to enable strategic and operational alignment, as defined by Ulmer (2011) and illustrated in the information system area by Ulmer et al. (2011). Our decision making process should hence enable a quicker convergence between the different people involved, and a reduction of the workload necessary for finding a satisfactory substitution product.

6.2 INTELLIGENCE PHASE

The Intelligence phase is the first activity of Simon's process (Simon, 1960). During this phase, the information about the problem must be gathered and criteria must be set in order to limit the number of possible alternatives. In our context of product substitution, the final result of this phase is a set of requirements that constrain in particular the mixture structure and the target properties to be matched. Concerning our CAPD tool, these requirements may concern specific chemical sub-structures, either to source the future molecules form renewable materials or to ensure their synthesizability.

6.2.1 Several business players

Our work performed in the framework of the project InBioSynSolv has highlighted that the setting of the parameters of our CAPD software requires several competences encountered in the chemical related enterprise: chemistry, thermodynamic, business and management. The project industrial partner defines the global specification, the chemical engineering expert sets the property models and targets and the chemist defines the chemical fields to explore. These persons are spread across the whole chemical supply chain as defined by (Marquardt et al., 2000) and presented in chapter 5 (Figure 71).

As we have seen in the previous chapter, the definition of the requirement can be considered as a decision making process itself (Aurum and Wohlin, 2003). As we are in an industrial context, we have chosen to use Ansoff (1965) and Anthony (1965) models. We thus distinguish three levels of decision making:

- The strategic decision making level where the project manager defines the business strategy and policy.

- The tactical decision making level where a business process expert is responsible of the respect of the business strategy.
- The operative decision making level where the constraints on the substitution product are set: the chemical engineering expert refines the property constraints and the chemical expert defines the constraints on the molecular structures.

We have hence four business players as represented on Figure 82.

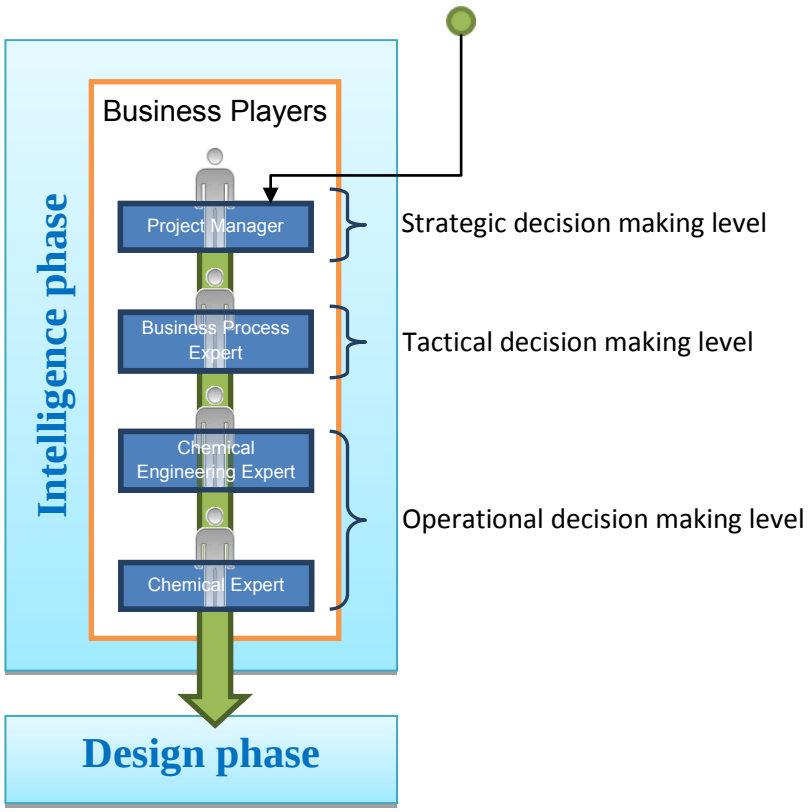


Figure 82: Business players in the Intelligence phase

A business player is a role: it has commitments that can be distributed among several people. Conversely, a single person can play several roles.

The project manager defines the business policy, including market opportunities and supply chain issues. The business process expert determines how the company objectives can be reached and he has a global overview of the supply chain. The chemical engineering expert is proficient to define the thermodynamic property constraints that the new molecule or product must satisfy. He also has in mind issues related to the chemical process context. The chemical expert brings information on the molecular system, like the available raw materials or the possible chemical transformations to synthesize the new molecules and to guarantee its chemical stability.

Concerning the operational decision making level, iterations between the chemical engineering expert and the chemical expert are mandatory. Indeed, even in a very simple case, the chemical engineering expert needs some information about the chemical structures chosen by the chemical expert in order to properly select the property estimation models. In the case where properties on a specific mixture component have to be set, a more complex iterative process must be performed.

6.2.2 *Modeling of the requirement*

For each business player, we use the concepts of model driven engineering to define the layers of abstraction of the information they are dealing with.

- The layer 0 corresponds to the real system i.e. to the knowledge as it truly exists, for example as documentation or simply in the player mind. This knowledge grasps every aspects of the enterprise, internal or external, at any level of detail.
- The model at layer 1 must show each one of the four views of enterprise modeling seen in the previous chapter: informational, organizational, functional and resource views.
- On layer 2, the metamodel used at each decision making level is specified.

The concepts of this approach are shown on Figure 83.

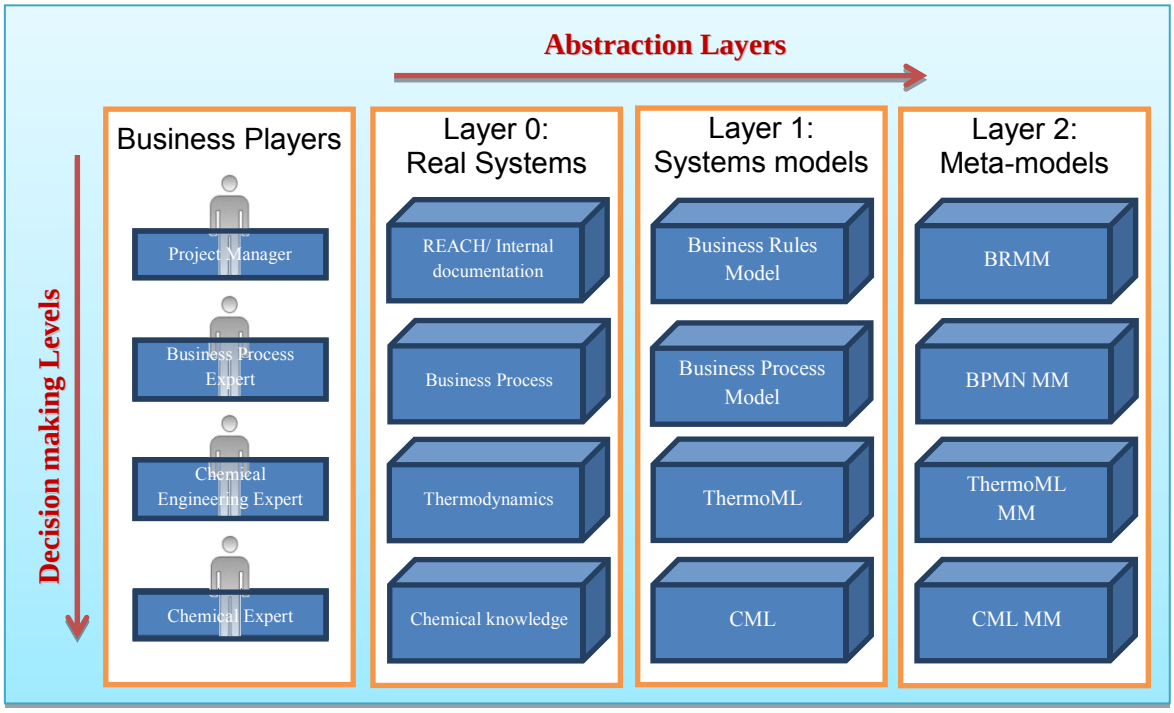


Figure 83: A global multi-level and multi-layer approach for the Intelligence phase from Heintz et al, (2012a)

The project manager deals with information such as the REACH regulation or internal documentation. This information can be modeled as business rules using a business rule metamodel (BRMN).

The business process expert manages the business process. He has to undertake actions in order to apply the business policy decided by the project manager while respecting of the business process already in place, such as the supply chain, the suppliers or the clients. His knowledge is represented as a business process diagram conformed to the Business Process Management Notation (BPMN) metamodel.

The chemical engineering expert decides about the constraints on the thermodynamic properties that the new product must meet, according to the requirements set at the previous levels and to his own knowledge of the chemical process where the new product should be used. His knowledge can be modeled as a file conformed to the ThermoML specification (metamodel) that are given in an XSD (XML Schema Definition) (Frenkel et al., 2006).

Then the chemical expert chooses the constraints on the molecular structure within the mixture product to find which are going to best respond to the requirements set earlier. He can define the number of elements if he chooses to consider a mixture, the synthons that a molecule should be synthesizable from or the composition of the different elements in the mixture. His knowledge can be represented as a file respecting the CML metamodel (Holliday et al., 2006). As ThermoML metamodel, CML metamodel is specified as an XSD.

With this information, the chemical engineering expert is able to associate the most suitable property estimation models to each property constraint. He also sets the constraints on the operating conditions that are necessary to run the chosen property estimation models.

One of the main issues of this approach is the operational alignment. Indeed, as each player uses his own model, it is necessary to align them in order to guaranty that each level understands fully the requirements set previously even if they have been specified with a different formalism.

As a first version, we have simplified this general approach. We have opted to model only the requirements, that is to say a part of the informational view of enterprise modeling. Moreover we have chosen to use only a unique model for all the decision levels and we also use an existing metamodel: UML2.

The Figure 84 represents the current approach.

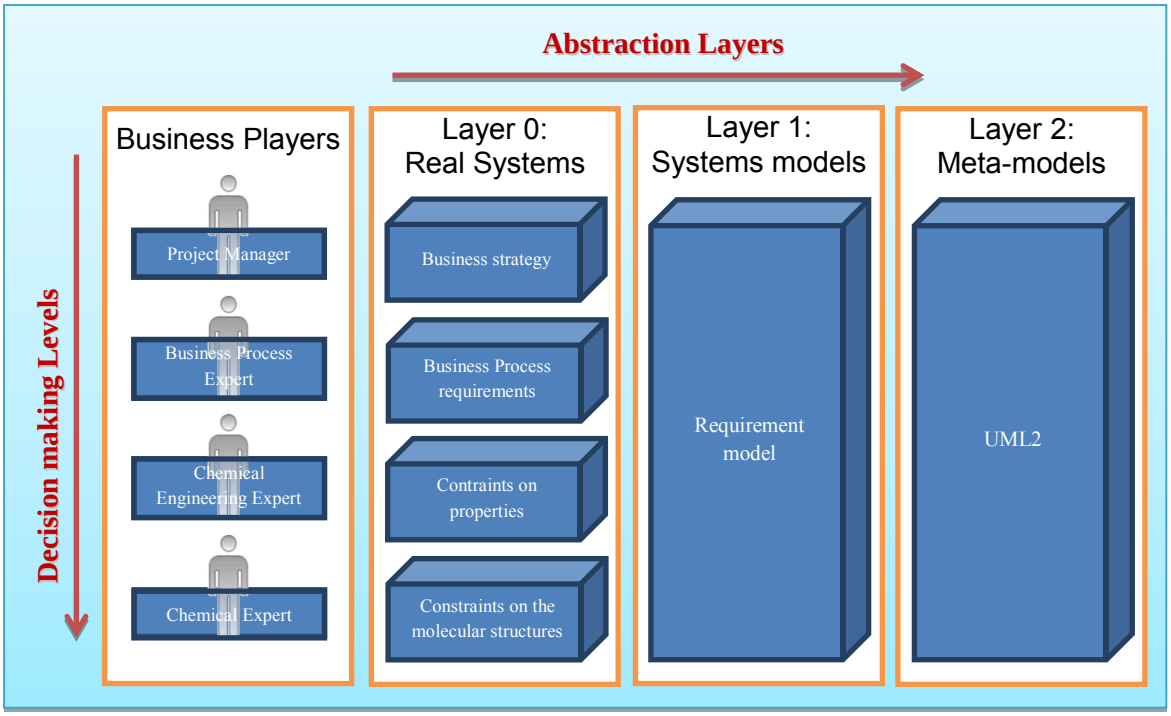


Figure 84: A simplified multi-level and multi-layer approach for the Intelligence phase

6.2.3 Model for requirements

Within SysML (modeling language dedicated to complex systems), there exists a part dedicated to requirements but we develop our own model based on UML2 diagram in a concern of consistency with the part B of this manuscript. Nevertheless, SysML requirement diagram could be used. The syntax would be different but the semantic would remain the same. We propose the requirement model given in Figure 85.

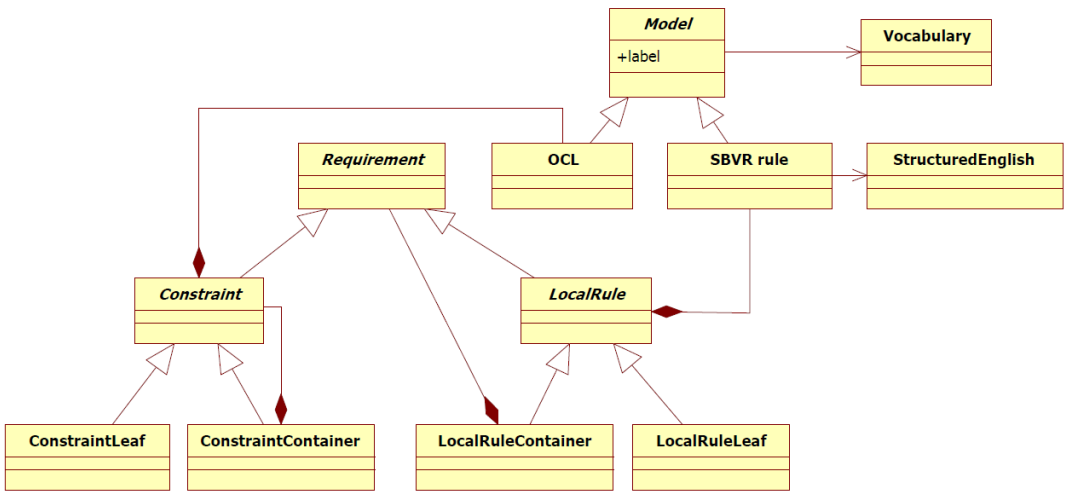


Figure 85: An UML model of requirement

Two types of requirement are available: constraints for the chemical engineering expert and the chemical expert and local rules for the project manager and the business process expert. The constraints are expressed using OCL and the local rules are expressed as SBVR rules using structured English.

It must be noted that traditional business rules are not used here as they are meant to constrain some aspects of the business and are applicable to the whole enterprise on a permanent basis. Indeed, for our application, we do not wish to define general company rules. Instead, we need rules specific to a given decision making process, and which must only last until the process is completed. That is why we have introduced what we call “local rules”. Local rules are temporary and apply only to the current decision making process.

Setting constraints with OCL also required some adjustments. Indeed, OCL is primarily dedicated to the setting of constraints on the classes (M1 layer). Just like business rules, these constraints are too general for our application. Indeed, we need to specify constraints on the objects which are instances of the class (M0 layer). It is possible to go around this problem by extensively using qualified associations in the associated class diagram. Qualified associations allow selecting the objects at the other end of the association. In addition, we need to define a property “type()” in order to define constraints on the type of constraint when there are inheritance relationships. It can also be noted that among the possible OCL conditions; invariant, pre- and post-conditions; we only need the invariant ones.

As represented on Figure 85, constraints and local rules can either be “leaf” or “container”. A container object will be refined by the next business players. They will associate it to new and more detailed requirements which integrate their own knowledge. A leaf object, on the other hand, will not be transmitted to the other players. At the end of the process, all container requirements must be associated to at least one more detailed requirement.

A local rule container can be associated to both local rules and constraints. On the other hand, a constraint container can only be associated to other constraints and not to local rules. In other words, constraints come from local rules and not the other way around. This is logical since, in our idea, local rules express higher level requirements than constraints.

6.2.4 Propagation of the decisions

A first issue concerns the management of all the information through the decision process. Using this model, the business players will sequentially create a requirements tree. An illustration of such a

requirements tree can be found in chapter 7. The tree will grow until the information it contains covers with sufficient details all the aspects necessary for launching the design phase.

As a triggering event of the decision process, a stimulus either conflicts with the business policy or motivates a change in the business policy. In other words, a stimulus can either violate an existing business rule or will lead to the addition or the modification of a business rule which will be violated. The decision of launching our decision making process is then made.

The local rule object associated to the violated business rule is created by the project manager. It is nevertheless possible that the solutions found to validate this rule will invalidate others. To avoid this to happen, the project manager has the possibility to add several local rules associated to business rules currently valid, but risking to be violated. After this step, the process is the same as for any other player.

For building the requirements tree, each player will sequentially derive the higher level requirements into more precise and specific ones. At each level, it is important for the business player to have access to the whole tree in its current state and not only to the requirements of the previous level. Indeed, when a requirement is derived from another one, it then includes more specific information in a particular domain, but the general information of the initial requirement might still be needed, in order to be derived in another domain. Such an awareness of all the available information participates to the strategic alignment that we aim at.

A second issue concerns the “classification” of the requirements. As the current business player uses most of the times the “container” requirements that are still associated to no other requirements, those later ones need to be highlighted. But on the other hand some requirements are not necessary and their presence may over the player with too much information.

In order to help a player to cope with the large number of requirements he may have to manage, every requirement has a level of interest. A requirement can be marked as “to be considered in priority”, “to be considered” or “ignored”. The “container” requirements that are still associated to no other requirements are automatically marked as “to be considered in priority”. The “leaf” requirements are automatically marked as “ignored”. All other requirements are by default marked as “to be considered”. Moreover, when the current player finds that a “container” requirement has been fully exploited and covered, he can set its level of interest to “ignored”. As the requirements “to be considered in priority” are highlighted and the requirements “ignored” are hidden, the requirements tree contains only the essential information for the next player.

A requirements tree is hence built by deriving the higher level requirements into lower level ones. Each time a business player creates a more specific requirement from a higher one, he implicitly makes a decision, which will later impact the global decision made at the end of the whole process. It is therefore important for each of these “small” decisions to be correct. With the approach proposed here, we ensure that, as the decision propagates through the levels of the enterprise, each business player has access to all the information he needs, while not being lost in useless details. The business players have thus a greater chance to make good decisions.

This propagation of the decision ensures a strategic alignment as the decisions taken have an impact on the lower levels. This way the business strategy is followed at each level of the enterprise.

6.2.5 The facilitator

As we have seen we propose a formalism to represent the local rules and constraints. Even though we have chosen to use rather accessible modeling languages for now, the different business players may need some help to correctly express their rules/constraints and may experience difficulties to understand the requirements set on the upper levels. A facilitator is thus needed. He is an expert of our approach and of SBVR rules and OCL constraints. He is in charge of monitoring the progress of the requirements tree building. He alerts the business players when action is needed from their side or when they are late in performing their actions. He must also be proficient in the use of the tool used during the design phase of the decision making process. Indeed, he will be in charge of translating the requirements in an understandable format for this tool in the Design phase.

A future perspective is to make this translation automatic. One can also consider making the monitoring of the progress of the Intelligence phase automatic. Both these perspectives will be permitted by the systematic use of formal languages for expressing the requirements.

6.3 DESIGN PHASE

The Design phase is the second phase of Simon’s decision making model. It mainly consists in the generation of the alternatives which are potential solutions for the problem requiring a decision. In our context, the alternatives are the different possibilities for the molecular substitution.

This part of the process can be schematized as presented on Figure 86 and is detailed in the following sections.

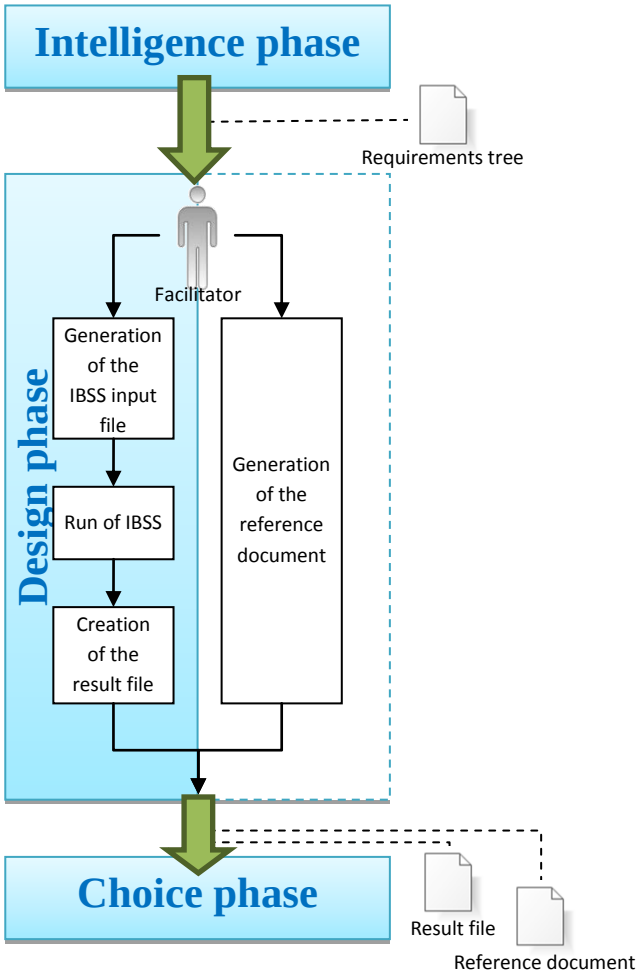


Figure 86: The Design phase

6.3.1 Generation of alternatives using IBSS

For generating alternatives, any method of molecular design can be used, even the trial and error ones. Indeed, using the same requirements tree, other means than our CAPD tool could potentially be used to generate alternatives. In other words, the requirements tree is not meant to contain the information focusing exclusively on the parameters of our tool. Now we consider the case where our CAPD tool, IBSS, is used.

For launching a research with our tool, the input file must be generated with the correct XML format, which sets the parameters of the objective function, of the mixture and of the genetic algorithm. The facilitator will use the requirements defined during the Intelligence phase to properly set this file. All the necessary data concerning the property targets and the constraints on the molecular structures is contained in the requirements tree. Indeed, since the facilitator also supervised the requirement setting, he is aware of any missing requirement during the Intelligence phase, and he has continuously requested the business players to provide them. However, the parameters of the search algorithm concern

specifically our CAPD tool. Setting these parameters requires being proficient in the use of our tool, which is not the case of the business players of the Intelligence phase. In addition, the facilitator is well indicated to set the parameters of the genetic algorithm.

It can be noted that some requirements of the requirements tree may no bet useful for the tool. Thus, they are not written in the CAPD tool XML input file but are kept in order to be used during the Choice phase.

Once the input file is finalized, the CAPD tool is run. It shall return products which are in theory reasonably good candidates for the substitution. But in practice their goodness depends on the accuracy of the property evaluation models currently available. Besides, purely theoretical molecular structures may not be actually feasible. Chemical synthesis feasibility rules could be integrated in our tool to reduce the probability to generate products which synthesis is impracticable. Finally, the product candidates depend on the weighting of the many constraints to be satisfied.

As a conclusion, it can be observed that the tool proposes candidate products supposed to be good ones, but that human expertise and laboratory consolidation are still necessary for various reasons. There is a need to select which candidates is the most promising one and worth being synthesized and tested in laboratory.

6.3.2 Generation of a reference document

In parallel of the generation of alternatives, the facilitator transforms the requirements tree into a document understandable for the experts involved in the Choice phase. This way the experts will be able to choose the product that best fits the requirements of the Intelligence phase. This document is a classic text document that translates the requirements tree. It will be used as a reference for the future steps of the decision process.

The document generation step does not traditionally belong to the design phase but, in our opinion, it is coherent to create such a document in parallel of the generation of the input file and the running of our CAPD tool. Indeed, during these steps, all the elements necessary for creating the reference document are available and such a document is mandatory for the Choice phase.

6.4 CHOICE PHASE

The Choice phase is the third and final phase of Simon's model. It consists in choosing which one of the alternatives generated during the Design phase will be implemented.

The inputs of this phase are the reference document explaining the requirements set at the Intelligence phase and the alternatives obtained as an output of the CAPD tool.

This part of the process can be schematized as presented on Figure 87 and is detailed in the following sections.

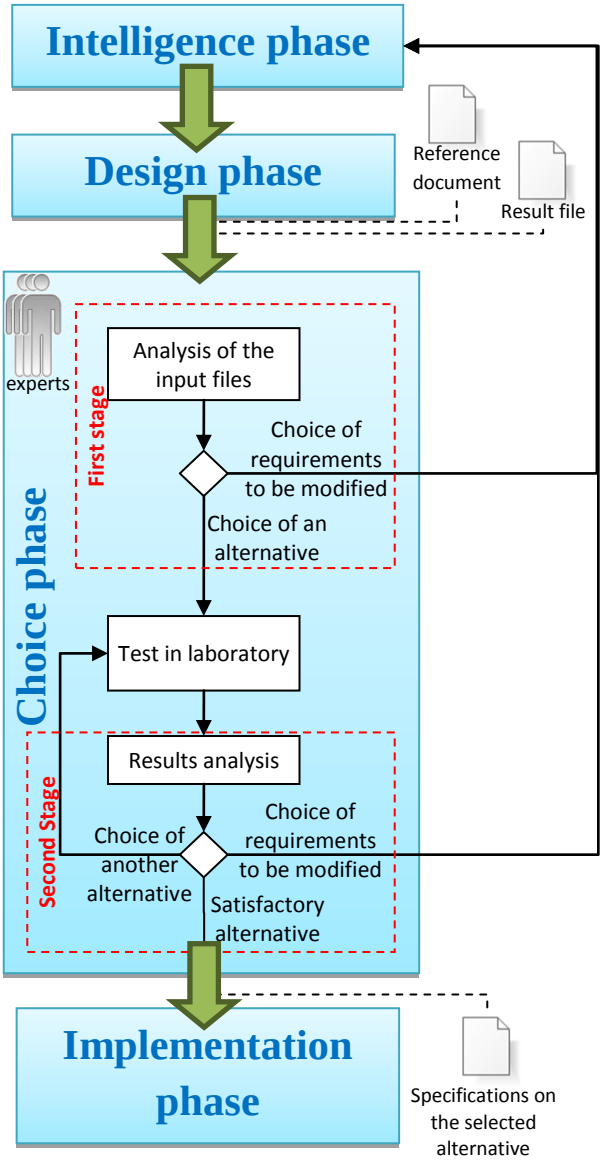


Figure 87: The Choice phase

6.4.1 Actors of the Choice phase

The actors of the Choice phase are the experts involved in the DELPHI method, introduced in chapter 5. Those experts must come from various domains in order for the final decision to be consistent

with the enterprise and economic environment. We stress that the business players of the Intelligence phase must not be members of the panel of experts in the Choice phase, in order to ensure impartial and supplementary judgments.

A proper selection of the experts' domain can help reducing the weaknesses of the results of our CAPD tool. Indeed, a chemist will identify unrealistic molecules and the alternatives containing such molecules will be discarded through the DELPHI method. In a same order of idea, a thermodynamic expert will recognize the misused models.

We have identified several domains that should be represented in the Choice phase. They concern chemical engineering (synthesizability, respect of the constraints), process system engineering (QHSE, adaptability of the constraints), industrial engineering (supply chain) and economics.

Finally, to guaranty a better efficiency of the DELPHI process, it is recommended that the experts must remain anonymous. Indeed, this allows putting each expert on an equal footing.

6.4.2 A two stage process

Before implementing any alternative, its properties must be known with an accuracy and a level of confidence that only laboratory tests can provide for molecules. As those tests can take several months and can be costly, it is thus necessary to carefully choose the alternatives that are worth testing. We have considered a two stage process inspired by the post-mortem phase of Howard method (Howard, 1966) presented in 5.3.2.2. First, the experts choose the best alternative on the basis of their knowledge and on the information given by the property estimation models used by the CAPD tool. This product is then synthesized and tested in laboratory. This phase corresponds to the gathering of information in Howard method (Howard, 1966), except it reduces the uncertainty directly on the actual performances of the alternative (the "profit" of this alternative), and not on the state variables of the problem. The experimental data then replace the property estimation and the experts choose if this alternative is still worthy in a second stage. If not, another alternative is chosen for testing. If on the contrary the experts are still satisfied with the alternative, it is decided to select this product for substitution.

At each one of the two stages through the answers given in the questionnaire, the experts can also decide that some requirements of the Intelligence phase must be modified. Under special circumstances, they may recommend to modify a very high level requirement that will reassess the need for a substitution

and that will have for consequences the end of the decision process. They hence perform a kind of double check of the requirements. Both stages are now described.

6.4.3 First stage: selection of an alternative for laboratory consolidation

The desired outcome of this first stage is the selection, for laboratory testing, of the most promising alternative. However other outcomes are possible if no alternative is considered promising. In this case, it can be decided that requirements of the Intelligence phase must be revisited. Four cases can be imagined:

- There is no alternative with a high enough performance. It can mean that the problem was too constrained and that some constraints, either on the property targets or on the molecular structures, must be relaxed. If the relaxation of those constraints is not an option then some higher level requirements (strategic or tactical) must be modified. This can lead to give up the decision of product substitution and put an end to the current decision making process.
- Some alternatives have a high performance but they are not credible from the experts' point of view. This can be the result of inappropriate requirements on molecular structure, property estimation models or operating conditions.
- The best proposed alternatives have medium performances except on specific properties. In this case, a better tuning of the weighting of the property target in the global objective function must be encouraged.
- In the case where promising alternatives exist, one must be selected for laboratory validations. A large panel of experts will enable to effectively filter the chemically non-feasible alternatives and select the best compromise on estimated performances for each property. It will also allow taking other factors into account such as the difficulty to synthesize, the cost of used raw materials and so on.

The questionnaires used for this phase are created specifically each time a Choice phase takes place, but are always organized as follows. For illustration, the questionnaire used for the case study presented in the next chapter is available in appendix 10.8. A set of questions guides each expert to determine whether he believes that the set of alternative is worthy of consideration or not. If not, the answers to the questions will help to understand why, and to identify a way forward. It is indeed important

to identify the wrong decisions of the previous phases and to know the corrective actions for the new Intelligence and Design phases.

Firstly, technical questions are asked closely linked to the requirements which have been previously set. These questions have been organized the following way in the questionnaire:

- The first group of questions concerns the way the performance has been calculated. The property considered, the targets fixed, the relative weighting of the properties set and the property estimation models used are submitted to the experts' appreciation. The experts also have the possibility to identify missing and/or unnecessary properties, and to give their opinion on the credibility of the returned property values.
- The second group of questions is focused on the way the structure of the mixture has been constrained. The experts give their opinion on the choice on the molecules, on the constraints that they must respect and on the chosen composition.
- The third group of questions is dedicated to the parameters set for the search algorithm.

These three groups of questions are limited to the various aspects on which constraints have been established. They hence follow the structure of the CAPD tool input file. However, in the case where some aspects have been forgotten, each group of question ends by an open question to which a free answer can be given.

Surely, the level of detail of these questions is quite high for some of the experts involved. For example, a marketing expert is not expected to know whether the models used for evaluating the property values are correct. Therefore, for each question, it is possible to simply answer "No opinion".

Secondly, following these groups of detailed questions dedicated to technical experts (but for which the opinion of everyone is welcome) the questionnaire becomes more general and accessible. Each expert is asked if he considers that the number of alternatives with a satisfactory theoretical performance is sufficient, and if he sees any other choices which could have led to obtain a better set of alternatives. Then, he indicates whether he considers that the set of alternatives proposed is worth consideration or not. If not, he details if a better set can be obtained, or if he believes that no satisfactory set can be obtained. In this latter case, it means that the expert believes that the substitution is not feasible.

In the case where the expert judges the set of alternatives to be worthy of consideration, he is asked to indicate the top 5 alternatives and the alternatives which have to be rejected. He also has to

justify his choices. For example, a chemical expert may identify that an alternative is not feasible, whereas all the other experts place it in their top 5. On the next round, those other experts will probably reconsider their top 5 knowing that this alternative will not be synthesizable. Therefore, knowing the alternatives to be rejected is just as important as knowing the top 5.

When a consensus is reached on a promising alternative, this product is synthesized and tested in laboratory. Then the second stage begins.

6.4.4 *Second stage: validation/invalidation of the alternative tested*

The second stage consists in evaluating if the chosen alternative is still satisfactory in the light of the experimental values returned by the laboratory testing. If it is not the case, another promising alternative must be tested. This goes on until a satisfactory alternative is found or until there are no promising alternative left for testing. In this last case, the requirements of the Intelligence phase must be reconsidered with a rational similar to the one of the first decision.

For illustration, a particular case where no promising alternatives would be left is the following. It might happen that, after having tested several alternatives, it appears that the one particular property performance is never close to the expected value. In this case, it can be assumed that the model used for evaluating this property was not correctly chosen, and that therefore all the alternatives currently available can no longer be trusted. It would then appear useless to keep on testing them in laboratory. This case is quite similar to the case considered in the first decision, where the performances are considered as not credible by the experts. It leads to the same outcome: requirements on molecular structure, property estimation models or operating conditions shall be reviewed.

The questionnaire used in this stage is very similar to the questionnaire of the first stage. A set of questions guides each expert to determine whether he believes that the alternative meets his expectations or not. If not, the answers to the questions will help to understand why, and to identify a way forward. In addition, if an expert is not satisfied with the alternative tested, he is asked if the set of alternatives currently available can still be used, in which case he has to indicate the top 5 alternatives and the alternatives which have to be rejected, just like he did during the first stage of the Choice phase. If in his opinion the whole set of alternative must be rejected, he has to mention if he considers that a better set can be obtained, or that no satisfactory set can be obtained.

6.5 GENERAL PICTURE OF OUR PROPOSAL FOR A PRODUCT SUBSTITUTION DECISION PROCESS

A general overview of the whole process is depicted in Figure 88.

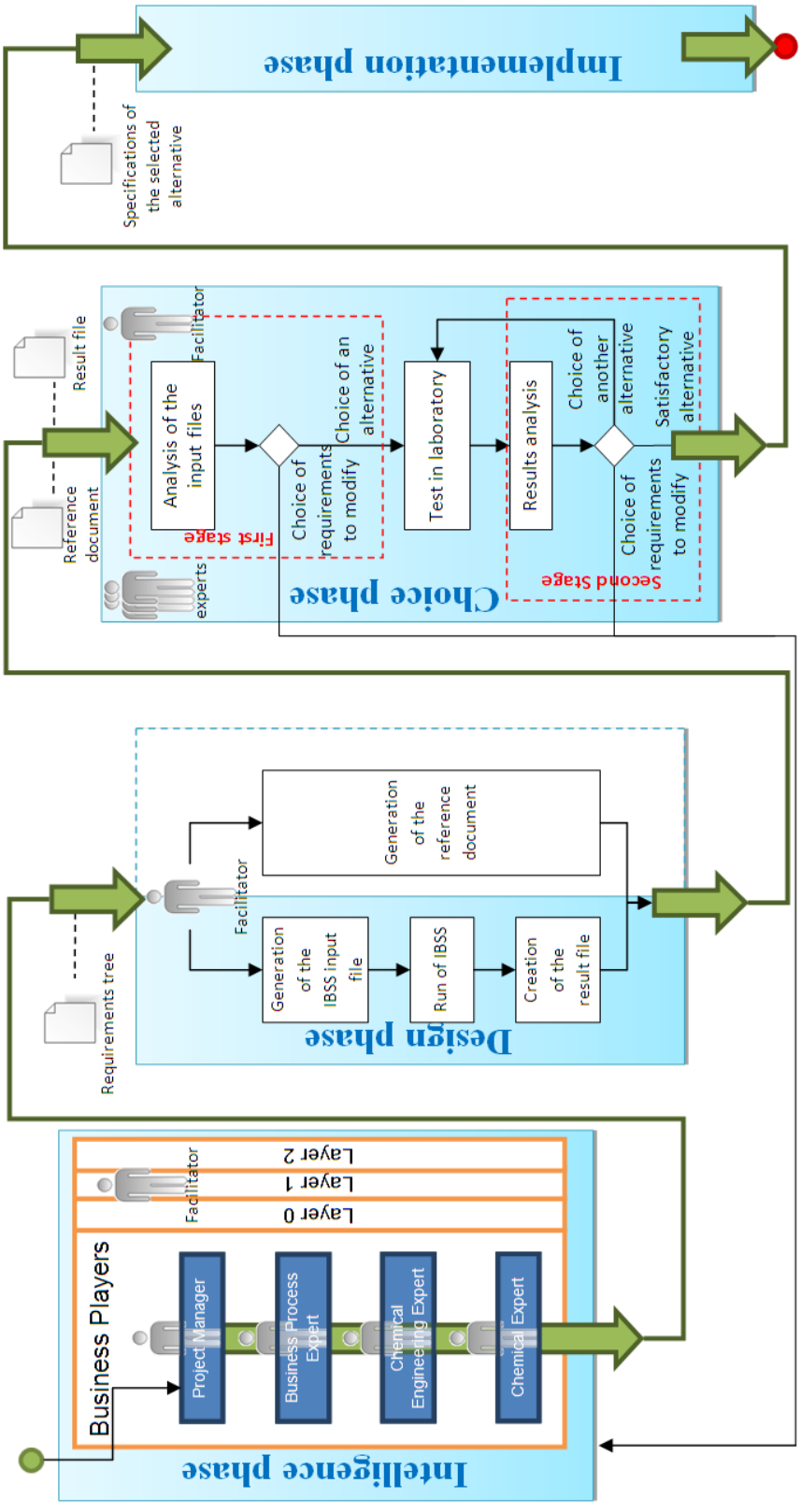


Figure 88: General picture of our decision making process

The fourth phase is the implementation phase that is not in our scope. It involves common process design activities. Its aim is to adapt the production process to the new product. Other decisions associated to preliminary design will then have to be made but they are not closely related to our decision process in the present state of our work.

6.6 CONCLUSION

In this chapter, we proposed a decision making process for the substitution of chemical products in an industrial context. This process follows the decision process of Simon (1960): namely the Intelligence, the Design and the Choice phases.

The Intelligence phase uses the concepts of enterprise modeling for ensuring a correct propagation of the decisions on the requirements through the different levels involved. Four business players spread across the whole chemical supply chain are involved. They use information formalized on different abstraction layers. The outcome of this phase is a requirements tree which translates the propagation of the decisions, and provides all the information necessary for the next phases. For building this tree, a modeling of the requirements is proposed based on a distinction between rules and constraints using SBVR and OCL formalism respectively.

During the Design phase, alternatives are generated by using our CAPD tool, once the appropriate input file is easily built with the requirements tree available with the help of a facilitator. In addition, during this phase, we propose to capitalize the information of the requirements tree in a reference document.

The Choice phase is divided in two stages inspired by the post-mortem phase of Howard (1966). It uses DELPHI methods where the questionnaires are built with technical questions related to the problem requirements relevancy and related to the chemical product alternatives pertinence. The first stage consists in involving a large panel of independent and anonymous experts in a DELPHI group decision method aiming at selecting the most promising alternative returned by the CAPD tool. This alternative is then tested in laboratory for determining its actual performance value. In the second stage, the experts are involved once again in a DELPHI method whose purpose is to decide whether the alternative tested is indeed satisfactory, or if another shall be tested. During both stages, it is possible for the experts to question the alternatives available, and consequently the requirements which have led to their generation. In such case, instead of going forward, the process goes back to the Intelligence phase.

The execution of the whole process relies on a facilitator who monitors the progress and provides assistance to the business players. However, since we extensively based our Intelligence and Design phases on computer approaches and model driven engineering, it can be considered that a large part of the tasks performed by the facilitator can be performed automatically.

CHAPTER 7

Industry related case study

In this chapter, through the combination of our CAPD tool and our decision making process, and with a case study based on Sinha and Achenie (2002), we illustrate how our approach can enable an efficient integration of environmental considerations among other enterprise needs. The case of a chemical company confronted to the necessity to find a greener solvent is treated. All the steps of our approach are detailed, and a focus is made on the information produced and on the role played by the enterprise resource, either human or material. The strategic, tactical and operational aspects of the problem are considered, and the propagation of the decision through these levels is highlighted. The possibility for our CAPD tool to be constrained by a wide range of enterprise requirements is demonstrated. The general frame retained for the final choice is detailed with more precision on several considerations which may arise in this particular case.

7.1 CONTEXT

In the lithographic printing process sector, ink residue and dried ink need to be removed from rubber blankets. The “blanket wash” is one of the most used solvents, and among the 40 formulations used in printing facilities throughout the United States, 21 contain petroleum distillates, raising environmental, health and safety (EHS) concerns (Sinha and Achenie, 2002).

The enterprise of our case study produces “blanket washes” and sells it to printing facilities. It also uses SBVR for expressing its business rules. In this context, we explain how our decision making process can be used for the substitution of the blanket wash.

7.2 STIMULUS

The whole process is triggered by a stimulus which claims for a decision to be made. In the scope of this case study, this stimulus is the following: pushed by regulation evolutions, and wishing to apply to the ISO 14001 standard certification about designing and implementing an effective environmental management system, one of the clients of the enterprise expresses the wish of having a greener “blanket wash”. As a consequence, a new fact is created in the business rule repository of the enterprise:

- Fact1: Customer C1 *wants* a replacement product that *is greener than* product Blanket Wash.

In addition, among others, the following business rules and fact already existed within the enterprise.

- Fact2: Each thing that *is wanted* by a customer *is* a customer need *of* that customer.
- BusinessRule1: It is obligatory that each customer need *of* each customer *is satisfied*.
- BusinessRule2: It is obligatory that each functionality that *is performed by* a product *is performed by* each replacement product *of* that product.
- BusinessRule3: It is obligatory that each supplier *is located at less than* 300 kilometers *from* the site *supplied by* this supplier.
- BusinessRule4: It is obligatory that each product *respects* the security rules.
- BusinessRule5: It is prohibited that a product *prevents* a process *to function*.

BusinessRule1 translates the commercial policy of the enterprise, which is to satisfy each need of its customers.

BusinessRule2 ensures that products are replaced by adequate substitutes. In this rule, “functionality” must be understood as what the product offers to the customer, and not as the chemical properties of the product. For example, a product removing ink can be used as a replacement product of the blanket wash, even if it does not have the same vapor pressure. Since the enterprises follows SBVR standard, the concept associated to the term “functionality” is explicitly defined, and any confusion is avoided.

BusinessRule3 is the result of a particular policy of the enterprise, which is to favor local suppliers in order to avoid the drawbacks of long distance transportation of goods (including the impact on the environment).

BusinessRule4 and BusinessRule5 are security and performance requirements.

It can be observed that the conjunction of Fact1 and Fact2 results in the violation of BusinessRule1. The business rules 2, 3, 4 and 5 are not violated but they are mentioned because they need to be considered since they may become violated by the substitution product.

7.3 INTELLIGENCE PHASE

As presented in the previous chapter, the Intelligence phase is the first phase of the decision making process. It mainly consists in the definition of the requirements by the business players thanks to a process guarantying the propagation of the decisions from the strategic to the operational decision making levels. To unambiguously express the requirements and as presented in chapter 5, SBVR Structured English is used on the strategic and tactical decision levels and OCL is used on the operational decision level. The facilitator assists the business players in expressing their requirements in these modeling languages and also helps them to understand the requirements set at the higher decision levels.

7.3.1 Strategic decision level

After having observed that BusinessRule1 is violated, the project manager launches the decision making process by starting the requirements tree. He defines the trunk of the tree by creating the following local rules:

- StrategicLocalRule1: **A replacement product *of* product Blanket Wash that *is greener than* product Blanket Wash must *be found*.**
- StrategicLocalRule2: **It is obligatory that each functionality that *is performed by* product Blanket Wash *is performed by* the replacement product *of* product Blanket Wash.**
- StrategicLocalRule3: **It is obligatory that each raw material *is available at less than* 300 kilometers *from* the production site.**
- StrategicLocalRule4: **The production cost *of* the replacement product *of* product Blanket Wash must *be at most* 10% *greater than* the production cost *of* product Blanket Wash.**
- StrategicLocalRule5: **It is obligatory that the replacement product *of* product Blanket Wash *respects* the security rules.**
- StrategicLocalRule6: **It is prohibited that the replacement product *of* product Blanket Wash *prevents* a process *to function*.**

StrategicLocalRule1 expresses the main decision to be made, for which the decision making process is launched. The strategic local rules 2, 3, 5 and 6 respectively come from the business rules 2, 3, 4 and 5, which the project manager has identified as risking to be violated. StrategicLocalRule4 translates a strategic decision of the project manager regarding the production costs objectives. A cost increase may be acceptable by the client at the printing facility since using greener product will help him getting the ISO 14001 certification.

The strategic local rules 1, 2, 3, 4, 5 and 6 are container rules. The requirements tree generated is represented on Figure 89. As explained in chapter 6, each requirement needs to be associated to a level of interest. The way the different levels of interest are represented on the tree is detailed in Figure 90.

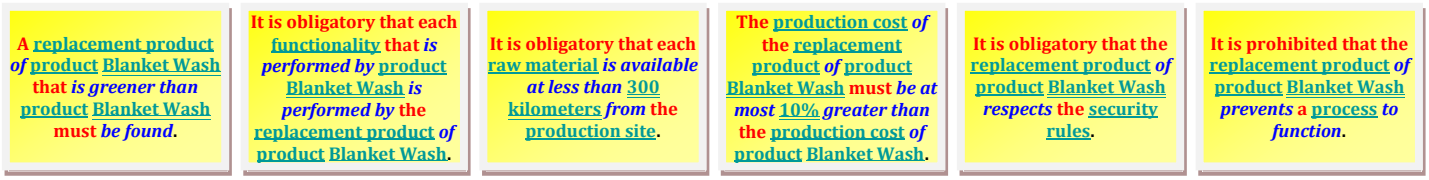


Figure 89: Requirements of the strategic decision making level

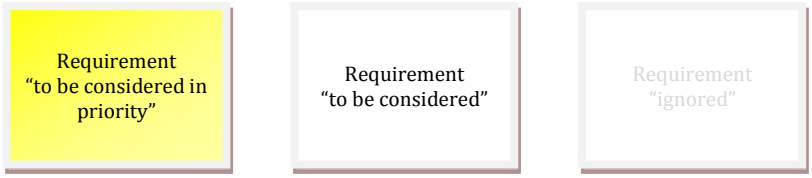


Figure 90: Formalism of the requirements tree

The project manager can now submit this trunk of requirements to the business process expert who will continue the Intelligence phase and complete the requirements tree.

7.3.2 Tactical decision level

The business process expert considers the first local rule and decides to refine it with:

- TacticalLocalRule1: **The replacement product *of* product Blanket Wash must *be* water based.**

This rule translates a tactical choice of the business process expert, which is to direct the research towards water-based solvents. Indeed, this expert believes that, with this orientation, the search will be quicker and the replacement product will have the greatest added value.

By taking into considerations supply chain issues, he also refines the strategic local rules 3 and 4 in:

- TacticalLocalRule2: **Raw materials *used for* replacement product *of* product Blanket Wash must *be available at* supplier S1 or *at* supplier S2 or *at* supplier S3.**
- TacticalLocalRule3: **The replacement product *of* product Blanket Wash must *be synthesizable with* the production means that *are available at* production site PS0.**

The suppliers S1, S2 and S3 are located at less than 300km from the production site PS0, and from the business process expert point of view, their costs are within acceptable limits with regards to the strategic objective of 10% increase of production cost at maximum set at the higher level. The production site PS0 is also judged compatible with this objective by the business process expert, provided that no additional investment on production means is required. This is summarized in the local rules above.

With these rules, the business process expert considers that he has fully exploited and covered the strategic local rule 3. Hence, he sets it as "ignored". This has for consequence that it no longer appears in the requirements tree for the following business players.

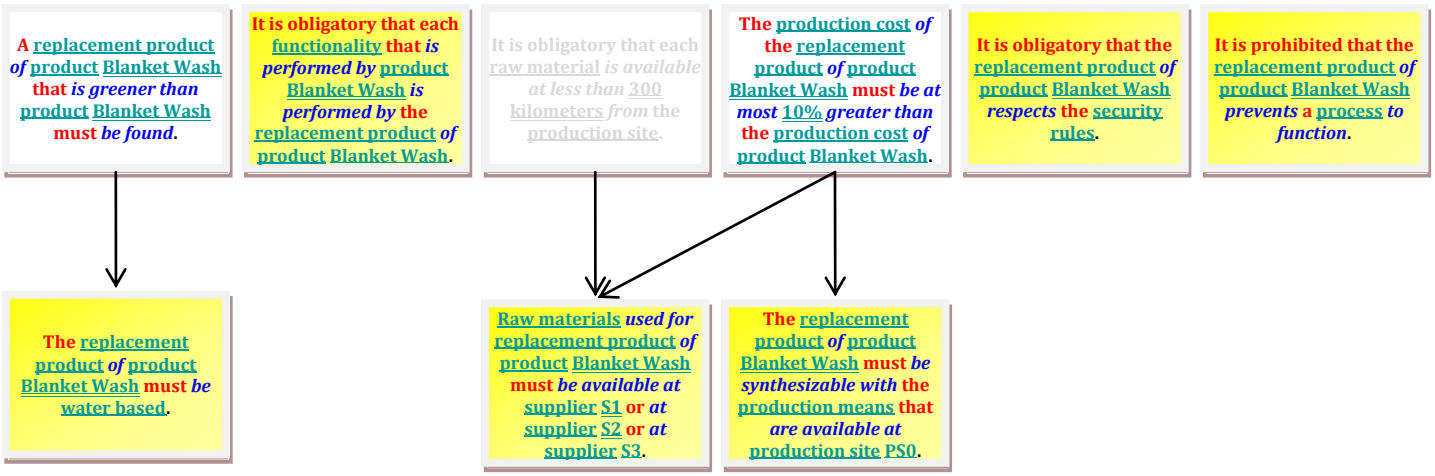


Figure 91: Requirements at the tactical decision making level

7.3.3 Operational decision level

The first business player to participate to the operational decisions is the chemical engineering expert who defines the properties and their targets for the new product. The chemical expert then sets the constraints on the molecular structure. Afterwards, the chemical engineering expert is involved a second time for defining which property estimation models are best suited.

7.3.3.1 Definition of the property targets

To define the property targets in OCL, the chemical engineering uses the following structure of domain entities (shown as a UML class diagram).

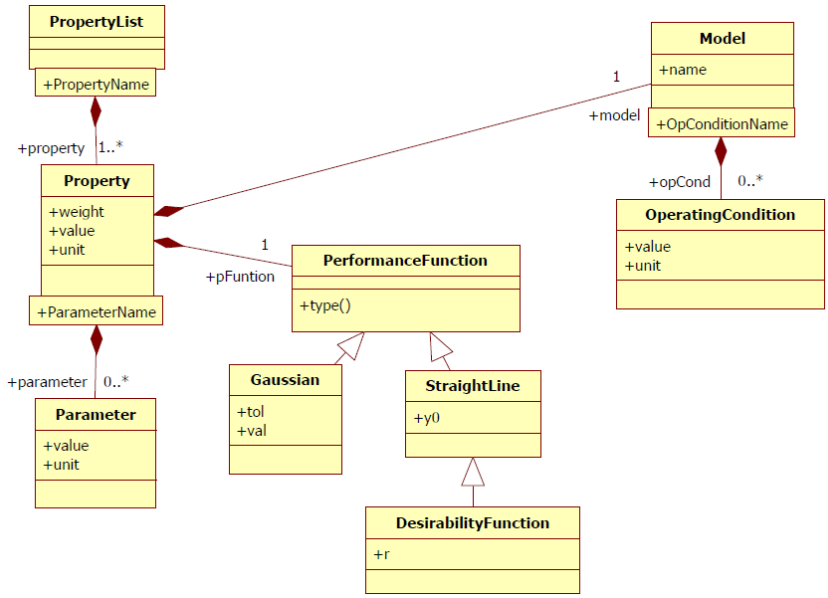


Figure 92: Class diagram used for the OCL description of the constraints on the properties

Among the requirements set previously, the chemical engineering expert first considers the requirements that are marked as to be considered in priority.

The strategic local rule 2 is considered first. This rule is focused on the functionalities that the product shall provide, in particular “solubililization” property. Since the product will be used for removing ink from rubber blankets, the chemical engineering expert refines the strategic local rule 2 by defining a target property associated to the solubility power of the future product on this specific ink. The ink is considered as a phenolic resin described by HSP parameter values used in setting the constraints as:

Context PropertyList **inv**
self.property[RED].weight=4
self.property[RED].value =0
self.property[RED].unit=”n/a”
self.property[RED].parameter[HSPd].value= 19.7
self.property[RED].parameter[HSPp].value= 11.6
self.property[RED].parameter[HSPh].value= 14.6
self.property[RED].parameter[HSPradius].value=12.7
self.property[RED].performanceFunction->type()=”Gaussian”
self.property[RED].performanceFunction.tol=1
self.property[RED].performanceFunction.val=0.8

We define RED as the distance in the Hansen solubility parameter space between the new product and the ink divided by the solubility radius of the ink. With this definition, a RED which is smaller than 1 is acceptable as it corresponds to the case where the new product is inside the solubility sphere. However, the smaller the distance between the two products is, the better the solubility is. Hence, the target of this property is 0.

Following the same formalism, several other property constraints are defined. The strategic local rule 5 stipulates that security rules must be respected. This “safety” requirement is refined in constraints on the flash point and the vapor pressure of the replacement product. Flash point is related to the explosion risk and the vapor pressure is related to the COV exposure of the operators. The strategic local rule 6 states that the replacement product must be compliant with the process in which it will be involved and the tactical local rule 3 specifies the production means from which the product must be synthesized. The chemical engineering expert refines them by defining constraints on the viscosity, the superficial tension and the density of the product. Finally the tactical local rule 1 that states that the mixture must be water-based is refined by the chemical engineering expert into a constraint on the water solubility (Log(Ws)) for all the components of the replacement product.

The tactical local rule 2 is not used by the chemical engineering expert and remains as “to be considered in priority”.

These new constraints are written using the same formalism as the one made explicit for the RED. The precise parameters are detailed in the following table.

Table 10: Parameters of the property constraints coming from priority requirements

Property name	Weighting	Target	Unit	Performance function type	Performance function parameters
RED (19.7;11.6;14.6;12.7)	4	=0	n/a	Gaussian	tol=1, val=0.8
Flash point	1	>323.15	K	Gaussian	tol=5, val=0.8
Vapor pressure	1	<0.00267	Bar	Gaussian	tol=0.0001, val=0.8
Viscosity (300K)	1	[0.8;1.4]	Cp	Gaussian	tol=0.1, val=0.8
Superficial Tension (298K)	1	[30;45]	dyn/cm2	Gaussian	tol=5, val=0.8
Density	1	[0.9;1.1]	n/a	Gaussian	tol=0.05, val=0.8
Log(Ws)	4	>4	mg/L	Gaussian	tol=0.5, val=0.8

Once these targets are defined, the requirements without priority are considered. The strategic local rule 1 stipulates that the new product must be greener. This is a rather vague property. The chemical engineer expert defined constraints on some environmental indices to account for the green features of the product. For illustration, the property on the Environmental impact indices is the following:

Context PropertyList **inv**
self.property[Environmental impact].weight=0.2
self.property[Environmental impact].value >8
self.property[Environmental impact].unit=”n/a”
self.property[Environmental impact].performanceFunction->type()=”Gaussian”
self.property[Environmental impact].performanceFunction.tol=1
self.property[Environmental impact].performanceFunction.val=0.8

The other property constraints related to environmental indices follow the same pattern.

The strategic local rule 4 which concern the price of the replacement product is refined in a constraint on the molecular weight. Indeed a heavy product is more likely to be costly to synthesize.

The parameters used for these constraints are given in Table 11.

Table 11: Parameters of the property constraints coming from non-priority requirements

Property name	Weighting	Target	Unit	Performance function type	Performance function parameters
Environmental impact	0.2	>8	n/a	Gaussian	tol=1, val=0.8
Environmental Waste	0.2	>8	n/a	Gaussian	tol=1, val=0.8
Health	0.2	>8	n/a	Gaussian	tol=1, val=0.8
Safety	0.2	>8	n/a	Gaussian	tol=1, val=0.8
LCA	0.2	>8	n/a	Gaussian	tol=1, val=0.8
Molecular weight	1	<200	g/mol	Gauss	tol=20, val=0.8

Once all the properties and their targets are set, the chemical engineering expert must weight them, as this is used to evaluate the performance of each alternative. As seen in chapter 2, the objective function is normalized by the sum of the weights of the properties. We propose to choose weights by following several principles:

1. The most important properties have the greatest weight. Here the RED property, about solubilizing the ink, and logWs ensuring a water compatible solvent are assigned a weight of 4.
2. Property estimation models which accuracy assessment for the case study is still under investigation are assigned a low weight. Here, the five indexes, Environmental impact, Environmental waste, Health, Safety, LCA, have not being truly validated, nor have a clear meaning; They are assigned an overall weight of 1, split into 5 times 0.2 for each index.
3. By default the weight is equal to unity.

A further comment on the assignment of weight is provided in section 7.3.3.3 about selection of the property models and their error.

All the constraints on the properties and their targets set by the chemical engineering expert are added in the requirements tree. For readability reasons, we have chosen to represent here only some significant parts of the tree.

Figure 93 presents a simplified version where only three property constraints are made explicit.

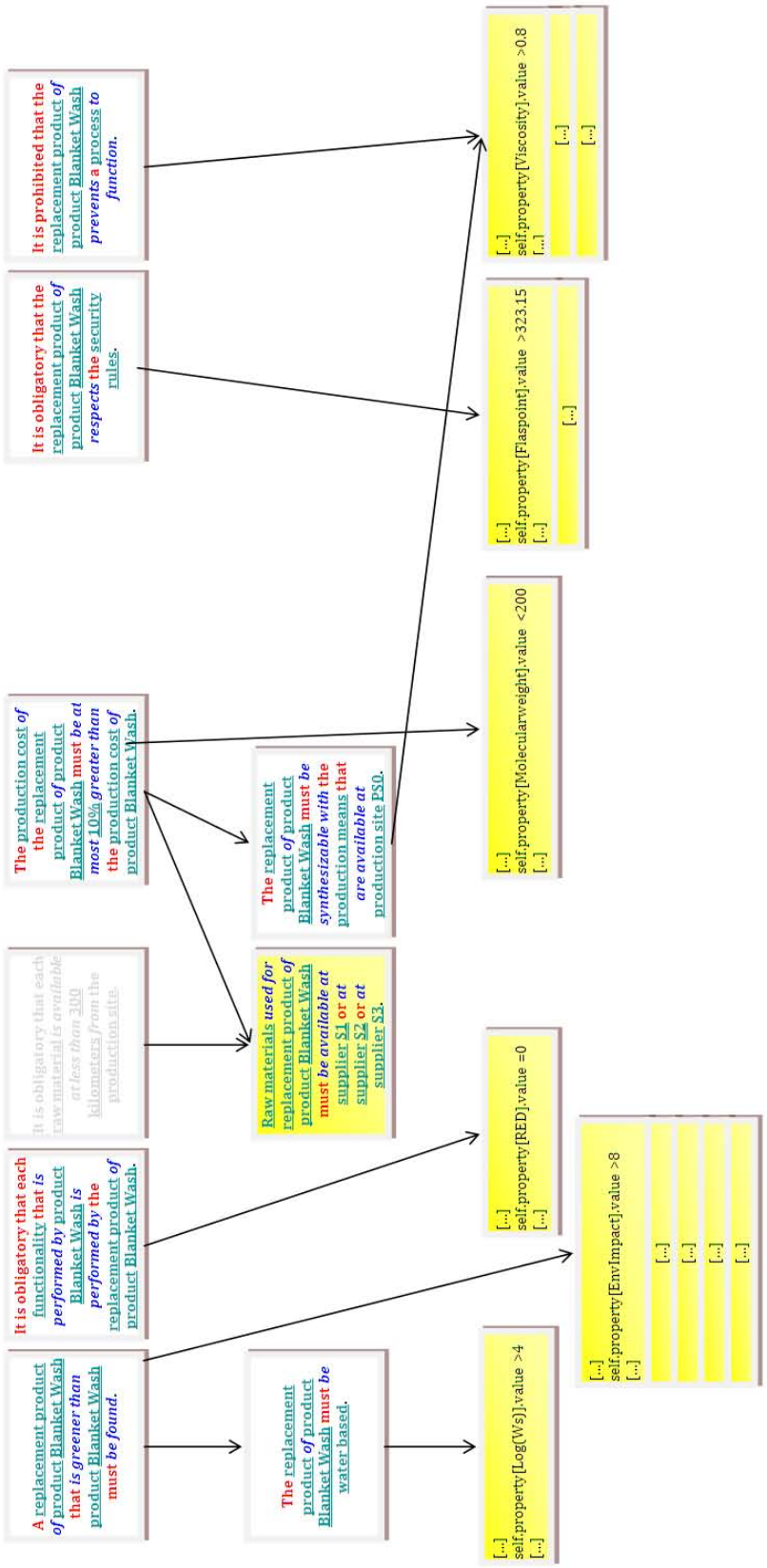


Figure 93: Partial view of requirements on the property targets

7.3.3.2 Definition of the molecular structure constraints

Once all the targets on the properties are available, the chemical expert is able to express adequate constraints on the molecular structure of the product. He uses the following structure of domain entities as a support of OCL:

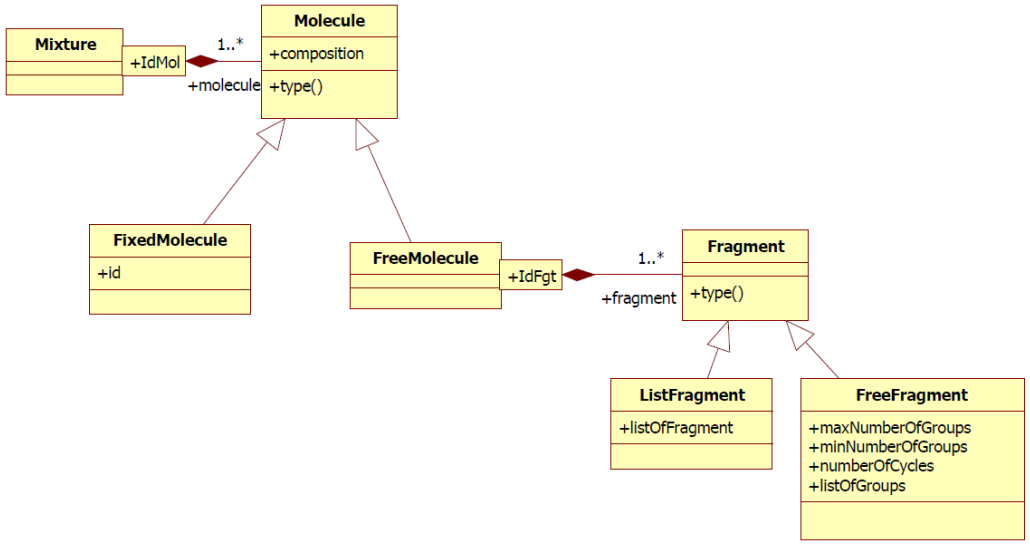


Figure 94: Class diagram used for the OCL description of the constraints on the molecular structures

First, the tactical local rule 1 is taken into account. This rule limits the search to water based products. The chemical expert refines it by constraining the product to be an aqueous binary mixture. Indeed, with knowledge of the property targets, he believes that a binary mixture is a promising solution. He thus writes:

Context Mixture **inv**
self.molecule->size()=2

Context Mixture **inv**
self.molecule[1]->type()=”FixedMolecule”
self.molecule[1].id=”water”

He also sets a constraint on the composition of the mixture, in order to have at least 30% of water in it:

Context Mixture **inv**
self.molecule[1].composition≥0.3

Since he considers that this requirement will not be useful for the other requirements remaining to be set, this constraint is created as a leaf in the requirements tree.

The binary mixture constraint implies that the structure of the second molecule in the mixture must be constrained. The chemical expert believes that the best choice is to deal with a free molecule made of

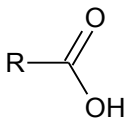
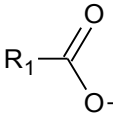
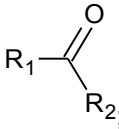
two fragments. The fragments are themselves constrained by taking into account all property target constraints set by the chemical engineering expert, as well as the strategic local rule 1 and the tactical local rule 2. As a reminder, the strategic local rule 1 states that the new product must be greener, while the tactical local rule 2 defines a set of suppliers where the raw materials must be available. Both these local rules restrain the fragments which can be used for building the molecule. As the chemical expert believes that a biomass based molecule would be promising, one fragment is a “ListFragment” containing biomass based synthons. For confidentiality reasons, those synthons are not made explicit.

Those constraints are formalized as followed:

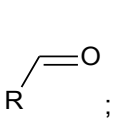
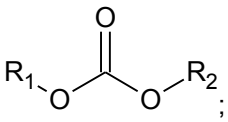
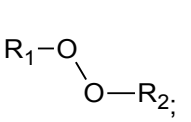
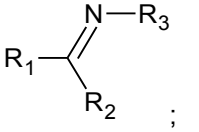
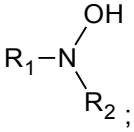
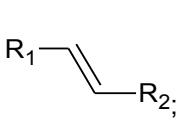
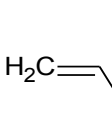
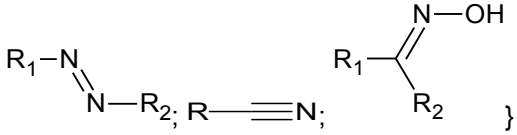
Context Mixture **inv**
self.molecule[2]->type()=”FreeMolecule”
self.molecule[2].fragment->size()=2

Context Mixture **inv**
self.molecule[2].fragment[1]->type()=”ListFragment”
self.molecule[2].fragment[1].listOfFragment={BMS1;…, BMS8}

Context Mixture **inv**
self.molecule[2].fragment[2]->type()=”FreeFragment”
self.molecule[2].fragment[2].maxNumberOfGroups=10
self.molecule[2].fragment[2].minNumberOfGroups=1
self.molecule[2].fragment[2].numberOfCycles=0
self.molecule[2].fragment[2].listOfGroups={-CH2-; >CH-; >C<; CH2=; -CH=; >C=; CH≡; -C≡;

=C=; NH2-; -NH-; >N-; NH=; -N=; N≡; OH-; -O-; O=;

}

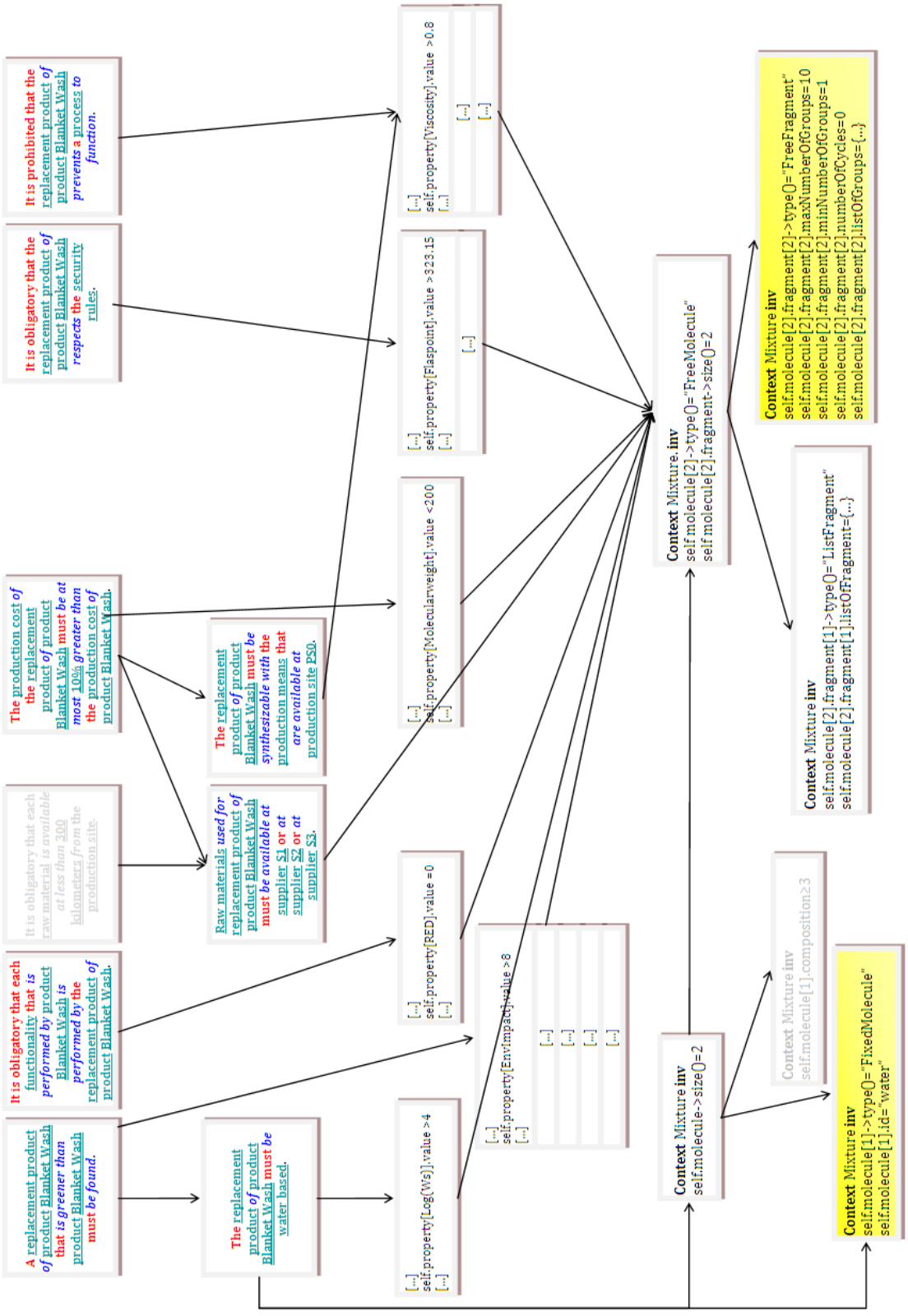


Figure 95 Partial view of requirements on the molecular structures

7.3.3.3 Definition of the property estimation models and the associated operating conditions

At this stage, all the constraints on the structure of the product to be found have been defined by the chemical expert. For each of the properties he previously identified, the chemical engineering expert has now the possibility to determine the best suited evaluation model and the associated operating conditions, and to translate this into constraints.

As the solution sought is a mixture, mixture property estimation models are selected. For aqueous mixture properties, linear dependence on molar fraction is assumed except for viscosity (model of Teja & Rice in Reid et al., 1987), surface tension (model of Tamura et al. in Reid et al., 1987) and Hansen parameters (dependent from molar volume).

For illustration, the models and operating conditions to estimate the Environmental impact, RED and Viscosity of each pure component within the mixture are:

Context PropertyList **inv**
self.property[Environmental impact].model.name=”Weis2009”
Context PropertyList **inv**
self.property[RED].model.name=”HSPiP”

Context PropertyList **inv**
self.property[Viscosity].model.name=”Conte2008”
self.property[Viscosity].model.opCond[Temperature].value=298.15
self.property[Viscosity].model.opCond [Temperature].unit=”K”

Following the same formalism, a property estimation model is associated to each property. The following table summarizes all the parameters used to set the constraints on the properties.

Table 12: Summary table of the properties, their target, their model and the associated parameters

Property name	Weighting	Target	Performance function	Mixture Model	Pure Compound Model	Operating conditions/model parameters
Molecular weight	1	<200 g/mol	G(20,0.8)			
Flash point	1	>323.15K	G(5,0.8)	linear	Catoire et al., 2006	
Vapor pressure	1	<0.00267 bar	G(10^{-4},0.8)	linear	Riedel, 1954	T=298.15K

Property name	Weighting	Target	Performance function	Mixture Model	Pure Compound Model	Operating conditions/model parameters
RED	4	<0	G(1,0.9)	Volumic fraction	HSPiP	HSPd=19.7 HSPp=11.6 HSPh=14.6 HSPradius=12.7
Env.Waste	0.2	>8	G(1,0.8)	linear	Weis, 2009	
Env.Impact	0.2	>8	G(1,0.8)	linear	Weis, 2009	
Health	0.2	>8	G(1,0.8)	linear	Weis, 2009	
Safety	0.2	>8	G(1,0.8)	linear	Weis, 2009	
LCA	0.2	>8	G(1,0.8)	linear	Weis, 2009	
Viscosity (300K)	1	[0.8;1.4] cp	G(0.1,0.8)	Tamura et al. in Reid et al., 87	Conte, 2008	T=298.15K
Superficial Tension (298K)	1	[30;45] dyn/cm2	G(5,0.8)	Teja & Rice in Reid et al., 87	Conte, 2008	
Density	1	[0.9;1.1]	G(0.05,0.8)	linear	HSPiP	
Log(Ws)	4	>4 mg/L	G(0.5,0.8)	linear	Marerro and Gani, 2002	

One must comment again the weight assigned when choosing the property themselves (section 7.3.3.1). Properties estimated with a model that gives a large error in the estimated value could be assigned a low weight. However, we do not recommend that. The first reason is that such property may be important for the problem and there would be a conflict between a low weight for large error and a high weight for importance. The second is that such error can be accounted for in the parameters of some of the performance functions, like the tolerance and value at tolerance of the Gaussian function.

At this stage, the Intelligence phase is completed. The output of this phase is the complete requirements tree, which contains all the constraints that the product to be found must satisfy.

7.4 DESIGN PHASE

Once the Intelligence phase is completed, the Design phase can start. The objective of this phase is to generate a set of alternative solutions satisfying the requirements expressed during the Intelligence

phase. In our scope, this phase mainly consists in the use of our CAPD tool, IBSS, and the creation of a reference document.

7.4.1 Use of IBSS

7.4.1.1 Input file

For running our CAPD tool, it is necessary to set its parameters regarding the objective function, the mixture and the genetic algorithm. This is done through an input file to which the CAPD tool will access when it is launched.

All the information for setting the parameters of the objective function and the mixture are available in the requirements tree built during the Intelligence phase.

The parameters of the genetic algorithm are defined by the facilitator, who will determine the best tuning of the CAPD search algorithm based on the choices made during the Intelligence phase. In our case study, based on several preliminary tests, these parameters are chosen as:

- Elitism: 30 individuals
- Population size: 100 individuals
- 300 iterations
- Probabilities of selection for modification:

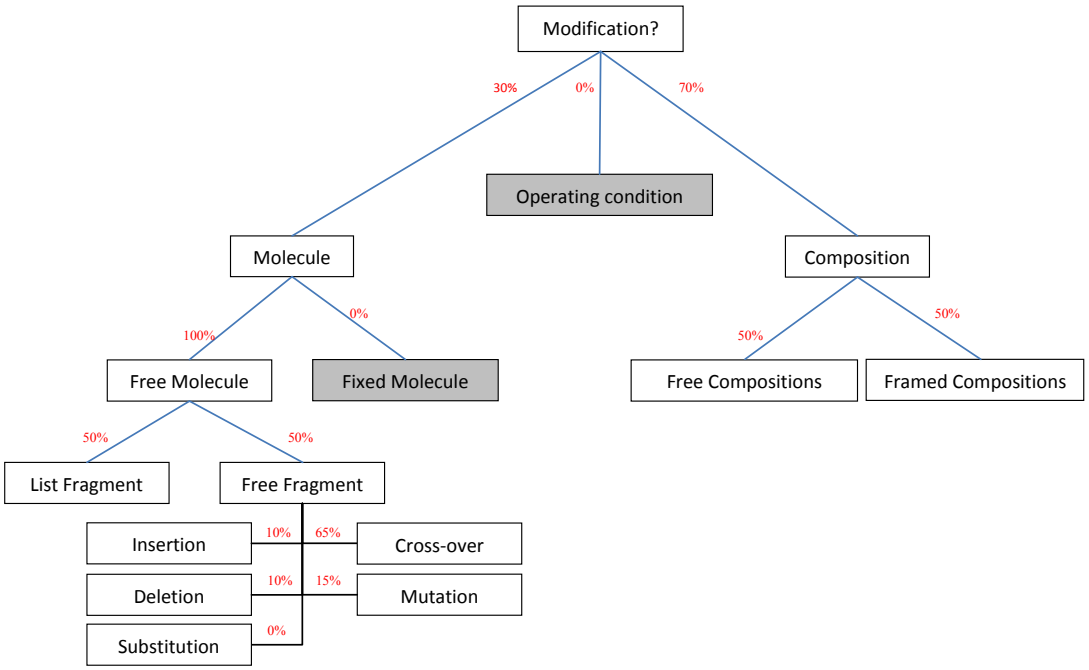


Figure 96: Distribution of the modification probabilities among the element defining a mixture

The facilitator then generates the input file of the tool by merging the information extracted from the requirements tree and his choices on the genetic algorithm parameters. The resulting XML file can be seen in appendix 10.6. It has to be noted that the translation of the requirements tree in an input file for our CAPD tool can be done automatically. This is particularly true for the OCL constraints, which are low-level requirements. However, for now, the facilitator supervision remains necessary.

The facilitator can now operate IBSS. The search is completed in less than 40 minutes. The results and the analysis of this search have been performed in (Heintz et al., 2012b).

7.4.1.2 Returned alternatives

The result file of the search presents each alternative with all the details on the mixture, the molecules and the composition, as well as the global performance of the alternative, the performance associated to each property and the property values calculated with the property estimation models.

The facilitator selects all the alternatives having a performance ranking higher than 95% of the highest alternative performance. This percentage is decided by the facilitator, but he can consult the chemical engineering expert of the Intelligence phase which is able to provide him the level of confidence of the property estimation models. The more the models are trustworthy, the fewer alternatives need to be taken into account in the next phase. The file resulting from this selection can be seen in appendix 10.7.

The results obtained demonstrate the performance of the genetic algorithm for our scope. Indeed, several solutions with very different molecular structures have been proposed, and the different possible compositions of a given mixture have been investigated. 45 mixtures are selected but, apart from water which was imposed, only 13 different molecules are proposed. Indeed, several mixtures are composed of identical molecules with a variation of the composition. This is in particular the case for the first 18 mixtures. The best mixture has a performance of almost 0.96 out of 1 and the last one to be selected has a performance of 0.94. There is hence a fair number of molecules with a performance which can be considered to be good.

It must also be noticed that all the 45 mixtures selected contain the same biomass based synthon as a ListFragment. Their FreeFragment, on the other hand, is more subject to variation. This unbalance between the variations of the ListFragment and of the FreeFragment is observed despite the fact that both have been assigned the same probability for modification in the search algorithm parameters. This

indicates that this specific biomass based synthon is probably the best from the list for satisfying the property targets.

7.4.2 Comparative analysis

The case study of the substitution of the “blanket wash” has been initially treated by Sinha and Achenie (2002). In their approach, they have selected 7 organic compounds, based on their water-solubility and their EHS impact, and they have then optimized the aqueous ratio with the help of an MINLP method. Their best result is a mixture of γ-butyrolactone and water (0.45/0.55), and according to our objective function, it has a performance of 0.94. This is a good solution but with a performance lower than our bests.

Sihna and Achenie (2003) relied on human expertise for selecting the solvents, and left the composition as the single optimization parameter for their search method. With our approach, we leave far more freedom to the computer tool, as it has already been presented. This is more computer intensive, but the field of exploration is much wider, and consequently, more innovative products may be obtained.

7.4.3 Reference document

In parallel to the generation of alternatives with our tool, the facilitator creates a reference document by transforming the requirements tree into a document understandable by the experts involved in the Choice phase. In this document, he also describes the parameters that he has set for the tuning of the CAPD search algorithm.

A document which will serve as a baseline for the future analyses is now available. It must be mentioned that this document also contains the leaf requirements which were hidden to some business players of the Intelligence phase. With this reference document and with the file describing the best alternatives returned, the Choice phase can begin.

7.5 CHOICE PHASE

The last phase of the decision making process is the Choice phase. The objective of this phase is to choose the best solution among the set of alternatives generated during the Design phase. In our scope, this phase is divided in two stages. The first one consists in identifying which alternative will be

consolidated by laboratory testing. The second one consists in determining whether the alternative is still satisfactory once the experimental results are available.

The experts use the documents provided by the Design phase, i.e. the list of the alternatives and the reference document that list all the requirements set during the Intelligence phase.

7.5.1 First stage

The first stage of the Choice phase consists in selecting the alternative which will be synthesized and tested in laboratory. Another possible outcome is the rejection of the set of alternatives available, leading to perform once again the Intelligence and the Design phases. This choice is made through a DELPHI method involving a large panel of experts. For this purpose, a questionnaire is written and submitted to the experts. The questionnaire corresponding to the first round is available in appendix 10.8. As already explained in chapter 6, the questionnaire is constructed as follows.

A set of technical questions closely linked to the requirements set during the Intelligence phase guides the experts to determine whether the set of alternative is worthy of consideration or not. If not, the answers to the questions will help to determine the corrective actions for the new Intelligence and Design phases.

The first technical questions concerns the way the performance has been calculated. They focus on the properties, the targets, the relative weighting of the properties and the property estimation models. The experts can also identify missing and/or unnecessary properties, and give their opinion on the credibility of the property values obtained.

For example, we can question the use of the five green indexes used to refine the “green” property. There are two reasons to consider their replacement by other properties and/or models. First, the estimation models of Weiss et al. (2009) used for these indexes were found unreliable in the InBioSynSolv project, as they did not predict that fluoride solvents were not green. Second, all these five indexes have been used without paying attention to an eventual duplication with other property models. Indeed, the safety index may duplicate with the flash point and the vapour pressure properties used to refine the “safety” properties associated with strategic local rule 5. For information, in more recent problems, these green indexes have been removed and replaced by properties with more physics, like the acute toxicity (property estimation model LC50), the biodegration factor BCF property estimation

model BCF) and the bioaccumulation factor (property estimation model Kow). New weighting were also considered.

Then the technical questions focus on the structure of the mixture. The experts give their opinion on the choice of a binary, water-based solvent, with a free molecule consisting of two fragments, one chosen in a list and the other constructed with given building blocks.

Afterwards, the technical questions are dedicated to the parameters set for the search algorithm.

Following these technical questions, the questionnaire becomes accessible to non-technical experts as well. Each expert is asked if the set of alternatives contains a sufficient number of alternatives with a satisfactory performance. Then, each expert mentions if the set of alternatives proposed is worthy of consideration, if a better set can be obtained, or if no satisfactory set can be obtained. In the latter case, it means that, according to this expert, the substitution is not feasible.

The experts who consider that the set of alternatives is worthy of consideration have to indicate the top 5 alternatives and the alternatives to be rejected.

For our case study, after each expert has answered this questionnaire, the answers are analysed and the questionnaire is updated in order to focus on the differences of opinion. A few other rounds follow, and a consensus is finally found.

For example, the experts considered that the issue of the unreliability of the five green indexes was not critical enough to stop the whole procedure because their total weight amounted to 5x0.2=1. They decided to inspect further the water-based mixture alternatives provided by the CAPD tool.

The mixture with the highest theoretical performance is retained for laboratory consolidation. This may appear to be a trivial choice, but it must be reminded that the first 18 alternatives (sorted by decreasing performance) consist of the same mixture with different compositions. This leads to believe that this mixture is both promising and robust to composition variation. Indeed, the figure below shows the evolution of the RED property with the fraction of the organic molecule occurring in the first 18 alternatives. The best alternative, with a fraction of 0.31, lies at the minimum of this curve. Then it has the best performance in RED, one of the most important properties considered, with a weight of 4, explaining in part why it is ranked at the top when considering the total performance.

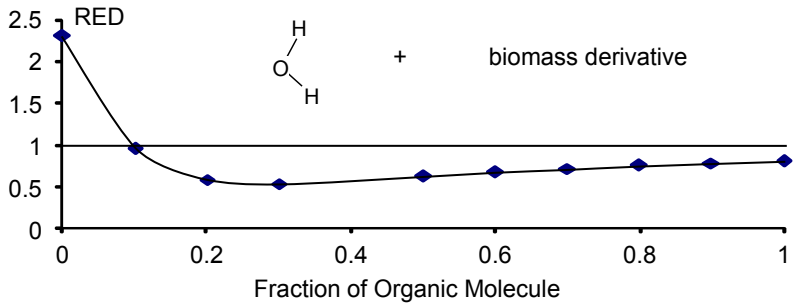


Figure 97: Best biomass based molecule and influence of its fraction on RED

Optimal Performance = 0.96; MW: 180.203 g/mol; FP: 356.45K; Pvap@298K: 7.47E-05bar; HSP_Distance: 6.72; RED: 0.53; Env. Waste_index: 8.13; Env. Impact_index: 5.48; Health_index: 7.74; Safety_index: 5.57; LCA_index: 4.37; Viscosity@300K: 1.41 cp; ST@298K: 44.64dyn/cm2; Log(Ws): 4.48mg/L; Density: 1.03; HSP_d: 16.75; HSP_p: 8.48; HSP_h: 15.04.

Since the product is water based, it is neither expensive nor complicated to perform laboratory testing on samples having different compositions. Hence, this opportunity is taken. With this perspective, the laboratory testing will in fact cover several alternatives at once.

As a conclusion of this first stage of the Choice phase, the chemical product with the highest theoretical performance is retained for laboratory consolidation, and opportunity to test samples having different compositions is taken.

7.5.2 Second stage

The second stage of the Choice Phase consists in determining if, in light of the laboratory results, the alternative tested is still considered satisfactory, or if another alternative from the set shall be tested. Similarly to the first stage, another possible outcome is the request to perform again Intelligence and the Design phases for obtaining a new set of alternatives. This choice is once again made through a DELPHI method involving the panel of experts of the first stage. For this purpose, a questionnaire is written and submitted to the experts. This questionnaire is constructed in a way close to the questionnaire of the first stage of the Choice phase.

As already explained in chapter 6, a set of questions guides each expert to determine whether the alternative meets his expectations or not. If not, the answers will help to identify a way forward. The experts who are not satisfied with the alternative tested are asked if another alternative can be selected in the set of alternatives currently available. Just like during the first stage of the Choice phase, the experts

who consider that another alternative can be selected have to indicate the top 5 alternatives and the alternatives to be rejected. The experts who reject the whole set of alternatives available must indicate if a better set can be obtained, or if no satisfactory set can be obtained.

Finally, one should comment that as experiments occur during the first and second stage and since the CAPD tool was ran based on a fully predictive approach, the molecules found may not be available on the market and may require synthesis. This might take some time. At that point, the cost of the molecules should also be considered by the experts in the light of the strategic local rule 4, allowing for 10% increase at most of the production costs.

Following the DELPHI process, the rounds will go on until a consensus is found. Since we do not actually have access to experimental results for the product we consider, we cannot go any further in the illustration of the decision making process without getting lost in conjectures. Nevertheless, with this case study, at this point, we have gone through all the phases of the process and the potential next stages are simply a repetition of what has already been presented.

7.6 CONCLUSION

In this chapter, we detailed a case study illustrating how our decision making process for the chemical product substitution is taking place in an industrial context. The case treated is inspired by one investigated by Sinha and Achenie (2002), i.e. the substitution of a solvent removing ink residue and dried ink from rubber blankets, and dedicated to the lithographic printing industry. We considered the case of a chemical related company producing this solvent and confronted to the request for a greener solvent by one of its client. With this case study, we have gone through all the phases of our decision making process.

In the Intelligence phase, the requirements on the product to be found are defined by business players from different levels of the enterprise. These requirements are defined by building a requirements tree, which grows bigger as the Intelligence phase progresses. On the strategic and tactical decision levels, SBVR Structured English formulations are used to unambiguously express a large variety of high level rules such as requirements on the performance of the product, on the supply chain or on security criteria. On the operational decision level, OCL is used to express lower level requirements expressing constraints on properties and on the molecular structures answering to the higher level rules, as for

example the definition of specific synthons in a molecule or the setting of targets on the vapor pressure or on environmental indices.

During the Intelligence phase, the strategic alignment is guaranteed as the decision propagates through the strategic, tactical and operational levels of the enterprise. For example, a strategic local rule related to a production cost objective is refined in a tactical level rule about the production site to be selected. This tactical local rule then leads to the definition of operational level constraints on the chemical properties of the product to be found in order for this product to be compliant with the production process.

In the Design phase, the requirements of the Intelligence phase are taken into account, in order to generate alternatives constrained by these requirements. The best alternative found by IBSS has a performance of 0.96 which is better than the mixture found by Sinha and Achenie (2002). The best alternatives proposed by our tool are selected and communicated to the experts of the next phase.

In the Choice phase, we have presented how the two stages are performed, and how the DELPHI method is used. At the first stage, the experts question the selection and weighting of some property estimation models but finally agree on testing the alternative with the higher theoretical performance. As we do not have access to the experimental results, the second phase has not been performed.

This chapter has demonstrated the adaptability skills of our CAPD tool, and has highlighted the interest of our decision making process in an industrial context involving a large number of persons with different expertise.

FOURTH PART

D. EPILOGUE

8.	Conclusion and Perspectives	……..	175
9.	Bibliography	……..	185
10.	Appendixes	……..	193

ABSTRACT

In this final part, a general conclusion presents the main outcomes and contributions of our work and is followed by a discussion on the limitations and perspectives.

The manuscript ends with the bibliography and with the appendixes.

CHAPTER 8

Conclusion and Perspectives

This chapter sets the conclusion of this PhD Thesis. The results are presented and the main scientific contributions are outlined. Finally perspectives are presented.

8.1 CONCLUSION

8.1.1 Context and work overview

Our research work has proposed a systemic approach aiming at enabling chemical engineering tools to handle sustainable and environmental issues. We have focused on the situation where chemical enterprises are confronted to the need to abandon some of their products and to replace them by “greener” ones. Despite the fact that this occurs frequently, stimulated by social awareness, environmental, health and safety related issues and a global reduction of resources, chemical companies are not equipped enough to face such repetitive challenges. Indeed, their traditional methods for answering these situations are time consuming, require consequent human and material resources, and are not always successful.

To answer this issue, we have developed and implemented a Computer Aided Product Design tool designed to be a credible alternative to the traditional methods for searching replacement products. In parallel, we have proposed a decision making process providing a guideline and a frame which will help the process of chemical product substitution to be performed efficiently, in an industrial context.

Although each can be used independently, we expect that the joint use of our CAPD tool and our decision making process will:

- ensure a correct propagation of the requirement decisions for product substitution through the different levels of the enterprise, in spite of the variety of the stakeholders involved, with needs that are sometimes diverging,
- propose rapidly a set of alternative solutions that are in coherence with all the needs and with the reality of the enterprise,
- enable to gather the relevant experts, to provide them the necessary information and to guide them to see the real products beyond the computer generated results, and thereby guarantee that the best alternative will be selected in the end.

In order to propose an approach which achieves these objectives, we explored several disciplines and retained techniques and concepts that were useful for our application.

8.1.2 Computer Aided Product Design in the chemical field

Analysis of the literature has highlighted that although computer aided molecular design is popular and studied in detail, computer aided mixture design remains rare and limited in scope. They all consist in finding molecules that satisfy a list of property related constraints set initially. Differences between the methods are numerous: e.g. on molecular representation models, on resolution methods and on performance criteria.

From the start, we have considered the design of a chemical product, either molecule or mixture, because it allows us to satisfy a considerable amount of constraints of diverse natures. By chemical product design, we cover the design of mixtures where the molecular structures, the composition and the operating conditions are investigated at the same time. To handle the numerous degrees of freedom that this choice set, we have used a meta-heuristic search genetic algorithm instead of enumerative techniques or exact methods, often used in CAMD methods published in the literature.

The genetic algorithm has been implemented in a multi-level framework to manage many types of property estimation mixture and pure component models and to improve the search efficiency: the complexity, and hopefully accuracy, of the models used increases at each new level, as the number of candidates decreases.

At each level, candidates are evaluated by aggregating their performance for each property target value. Each property performance can be customized by selecting various functions among which the Gaussian function is suitable to account for the model accuracy and confidence.

We have chosen to use the molecular graph representation of Korichi et al. (2008) to which we have added information on the bond type. This representation can manage any molecular structure, even cycles, while remaining computable and understandable by users. Ambiguity is also avoided and the graph chemical properties can be estimated with a greater precision. Besides this representation is fit for the genetic algorithm and can generate inputs for many types of property estimation models.

Such a detailed representation is compliant with the variety of needs of chemical related industry, like the ability to constrain part of the chemical product mixture formulation or part of each molecule structure or part of fragments within each molecule, by assigning fixed, list or free attributes to these variables. This enables the search of molecules sourced from renewable resources: molecules, synthons,

chemical function blocks. The genetic algorithm has been adapted to handle all this flexibility, in particular the modification operators like the possibility of inserting/deleting whole branches in the molecule.

8.1.3 The Computer Aided Product Design tool

The computer aided product design method has been implemented in the IBSS tool. The development of this tool has been model driven and its architecture is component and object oriented. The IBSS tool development relies upon four components: a search algorithm written in C#, a property calculation part written in VB.NET, a man-machine interface written in java and a functional group database stored as an XML file. Prior coding, UML2 has enabled to identify the users (basic and experts) and the main software functions listed afterwards. BPMN diagrams have described some dynamic behaviors between and within the software packages.

To ease the use of the tool by basic users, we have defined in parallel to a ‘calculable’ type of property, a ‘real’ type of property that describes product qualities, like “safe”, “toxic”, “volatile”. The qualitative scale of each ‘real’ property is associated with the quantitative scale of one or more ‘calculable’ property defined by the expert user.

The component and object oriented development eases the maintenance of the tool and enables to use the components independently. The tool has been tested and validated through several academic and industrial case studies.

8.1.4 A decision making process for chemical product substitution

A product substitution is generally performed at an industrial scale. However, in an industrial context, the issues at stake are numerous. Therein lies a large part of the complexity of a product substitution. A systemic approach is therefore needed to formalize a decision process for the chemical product substitution in an enterprise context.

Our decision process has been inspired by Simon (1960) whose three phases, Intelligence phase, Design phase and Choice phase, have been adapted to our problem.

During the Intelligence phase, the informational view, the resource view and the functional view of the enterprise have been taken into account to fully describe the requirements on the new product. Four business players coming from different decision making levels as defined by Ansoff (1965) and by Anthony (1965) have been introduced. On the strategic decision making level, the project manager

decides the business policy of the enterprise. On the tactical decision making level, the business process expert sets requirements so that the supply chain policy is respected. On the operational decision making level, the chemical engineering expert and the chemical expert transform the requirements into concrete constraints on the chemical product to find.

In order to ensure the alignment of the requirements through the enterprise, a model driven approach has been used and a simplified model of requirements based on UML2 has been proposed. In this model, high level requirements are expressed thanks to local rules, inspired by business rules and using SBVR Structured English. The lower level requirements are expressed as constraints written with OCL.

Starting from the strategic decision making level and helped by a facilitator, the business players are sequentially asked to refine the requirements on the basis of their knowledge and experience. Progressively, a requirements tree is created, in which the trunk and the main branches correspond to the high-level requirements, while the small branches correspond to the low-level ones. At the end of the process, the tree contains specific requirements on the replacement product, such as a vapor pressure target or a constraint on the use of specific synthons.

The Design phase consists in using this information to generate promising alternatives. In this work we have used IBSS, our CAPD tool. The facilitator is in charge of building the input file with the information available in the requirements tree. He is also responsible for tuning the search algorithm. When the input file is complete, the tool is launched. A percentage of the best alternatives are selected to be used in the Choice phase. In parallel, the facilitator translates the requirements tree into a reference document. In this phase, many activities are performed by the facilitator. However, a large part of these activities can also be run automatically.

During the Choice phase, the best alternatives of the Design phase are analyzed in light of the reference document. When the alternatives are generated with a computer aided design method, as it is the case with our approach, the margin of error due to the use of property estimation models imposes a consolidation by laboratory testing, which is potentially long and costly. The Choice phase has thus been divided in two stages. In the first stage, a specific alternative is chosen for laboratory testing. In the second stage, the laboratory results are analyzed in order to decide if the alternative is indeed satisfactory. At both stages, a DELPHI group decision-making method is used. The experts involved in

the DELPHI processes are independent and anonymous and remain the same for both stages. If they are not fully satisfied with the proposed alternatives, the experts have the possibility to request the modification of some requirements. In such cases, the Intelligence phase and the Design phase are run once again so that a new set of alternatives is proposed.

8.2 SYNTHESIS OF SCIENTIFIC CONTRIBUTIONS

The main contributions of our work are highlighted hereafter.

- Application of concepts of system engineering for the development of an approach for sustainable growth that takes into account every aspect from the molecule to the enterprise scale: that is to say molecule, chemical product, chemical process, business process, enterprise.
- Multi-disciplinary work at the interface of several disciplines: chemistry, chemical engineering, industrial engineering and software design.
- Proposition of a new method of Computer Aided Product Design:
    - Simultaneous investigation of molecule structures, composition, and operating conditions ,
    - Adaptation of the genetic algorithm to product design,
    - Addition of the possibility to constrain molecular fragment, enabling the search of molecules sourced from renewable resources.
- Development of a CAPD software tool:
    - Implementation of an object and component oriented architecture,
    - Adaptation of the genetic algorithm modification operators to molecular graphs.
- Proposition of a structuring frame for a decision making process for sustainability in chemical engineering:
    - Implementation of a frame enabling the strategic alignment of the requirements,
    - Proposition of a model of requirements including business rules, constraints and modeling languages,
    - Proposition of a two stage Choice phase for integrating laboratory testing.
- Close collaboration with an industrial partner and partial validation of our work by a third party

8.3 LIMITATIONS AND PERSPECTIVES

8.3.1 Computer Aided Product Design method

- Our CAPD method and the tool we have developed are currently limited in their application to solvent design. An interesting perspective would be to extend their application to the design of other chemical products, such as vegetable oils or polymers for example. This implies that suitable property estimation models must be integrated in the property calculation library. For polymers, the molecular representation must also be adapted, and the search algorithm must be modified to take into account this new representation. Generally, we have to face our domain specific CAPD method to CAPD concepts coming from manufacturing field.
- As of today, no feasibility rule, like rules checking the chemical synthesis easiness, is implemented in our CAPD tool. Implementing these rules would enable to perform a selection among the candidate solutions generated during the search by penalizing the ones that are not realistic. For achieving this objective, a work with our chemist partners is necessary to identify some satisfactory feasibility rules, that is to say whose range of application is wide enough, and for which no or very few exceptions exist.
- Another perspective for our tool is to propose several search algorithms. Meta-heuristic methods such as simulated annealing or Tabu search can easily be adapted and integrated thanks to our software architecture. Indeed the model driven conception and the object and component oriented architecture make the implementation of all these perspectives feasible.
- One last perspective is to integrate some property estimation models related to properties of industrial interest. For example a method able to predict the cost of a product would be useful in an industrial context, although building such a prediction model is a complicated task.

8.3.2 Decision making process

- We have detailed our decision making process up to the Choice phase as in Simon's process. A future perspective is to consider the Implementation phase as well as in Vallin's process. Indeed in this phase, some issues may arise and lead to reconsider the outcomes of the Choice phase or of the Intelligence phase. For example, it may happen that during the Implementation phase, the

substitution product purchase costs and the production process costs (due to the integration of the new product in process) appear to be greater than expected.

- We have reduced our approach by selecting a unique model for the requirements and imposing it to all the business players. The next step is to handle the models used by each business player and to guarantee an alignment between all these models taking benefit from models transformation technics.
- Each phase of the approach we propose relies on a facilitator. A large part of his tasks can be performed automatically. A future perspective is hence to investigate further this automation. Examples of tasks which can be automated are the monitoring of the progress of the Intelligence phase and the translation of the requirements tree into an input file for the CAPD tool. Some tasks can also be partially automated, such as the creation of the reference document gathering the requirements and the writing of the questionnaires which will be used in the DELPHI rounds of the Choice phase.
- Our approach should also be supported by computational tools. For this purpose, existing tools must be chosen and adapted and/or tools must be developed for supporting each step of our process. Applications coming from BRM, DM, MDE, enterprise and information system engineering have to be considered.
- Another perspective is to enable the capitalization of the activities performed during the Intelligence phase. Indeed, it would be interesting to keep track of the reaction adopted in front of a given situation, so that similar situations can be treated more rapidly. In practice, it means that it should be possible to store some complete parts of the requirements tree. For illustration, let us assume that a part of the requirements tree is stored, which contains the development of a requirement such as “product XX must be replaced by a greener one” into constraints (on the environmental impact, environmental waste, etc.). It would then be possible to reuse this part of the tree as a template in a situation where a new requirement would be “product YY must be replaced by a greener one”.
- More generally that would be a high benefit to work on Knowledge Based Engineering (KBE) (Bodein et al., 2012), especially in the Intelligence phase but also in the whole frame. Our proposal relies on skill and expertise from several human players. One of them, the facilitator, is a main actor for the success of the complete process. This can be seen as a limitation and a true drawback. To think on concepts and tools from KBE would be surely a strong progress line. Coupled to this scientific concern, innovation based methods should be furthermore analyzed. Semantic web technologies can also complete a KBE approach.

- On the way to compute the whole process, information communication technologies can bring many benefits in order to manage the Intelligence and choice phases more efficiently. Particularly technologies from Web 2.0 issues can ease the facilitator and experts' tasks and players' collaboration. The different actors (players, experts and facilitator) can have different locations. Those technologies would allow managing remote activities and formalize a stronger collaborative decision making. The Delphi method can then be replaced by another expert-based method in order to improve the consensus getting thanks to collaboration supporting tool.
- Finally, we have simplified our approach by considering that the persons involved in our decision making process were "perfect", that is to say competent, motivated and meticulous. Our process can therefore be improved by taking into account the social and psychological aspects. For example, methods for motivating peoples and methods for avoiding withholding of information can be integrated in our approach.

CHAPTER 9

Bibliography

Aarts, E., van Laarhoven, P., 1985. Statistical Cooling: A General Approach to Combinatorial Optimization Problems. Philips Journal of Research 40, 193–226.

Achenie, L.E.K., Gani, R., Venkatasubramanian, V., 2003. Computer Aided Molecular Design: Theory and Practice. Elsevier.

Ansoff, H.I., 1965. Corporate strategy an analytic approach to business policy for growth and expansion. McGraw-Hill, New York.

Anthony, R.N., 1965. Planning and control systems a framework for analysis, Studies in management control. Division of research graduate school of business administration Harvard university, Boston.

Ashrafi, N., 1998. A decision making framework for software total quality management. International Journal of Technology Management 16, 532–543.

Aurum, A., Wohlin, C., 2003. The fundamental nature of requirements engineering activities as a decision-making process. Information and Software Technology 45, 945–954.

Baez Senties, O., Azzaro-Pantel, C., Pibouleau, L., Domenech, S., 2010. Multiobjective scheduling for semiconductor manufacturing plants. Computers & Chemical Engineering 34, 555–566.

Baudouin, O., Dechelotte, S., Guittard, P., Vacher, A., 2008. Simulis® Thermodynamics: an open framework for users and developers, in: 18th European Symposium on Computer Aided Process Engineering. Elsevier, pp. 635–640.

Belaud, J.-P., 2009. System Modeling & Analysis (SMA): Systeme Modeling with Unified Modeling Language 2, cours ISAE-SUPAERO master System Engineering.

Bézivin, J., 2004. Sur les principes de base de l'ingénierie des modèles. RSTI-L'Objet 10, 145–157.

Bodein, Y., Rose, B., Caillaud, E., 2012. Decisional Model for KBE Implementation in a Commercial CAD Software. Computer-Aided Design and Applications 9, 121–131.

Booch, G., Rumbaugh, J., Jacobson, I., 2000. Le Guide de l'utilisateur UML. Eyrolles.

Boucher, X., 2007. Vers un pilotage agile de lévolution des sysèmes de production. Habilitation à diriger des recherches, Ecole Nationale Supérierue des Mines de Saint-Etienne.

BRG, 2000. Defining Business Rules - What Are They Really? [WWW Document]. URL http://www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf

Briggs, T., Kunsch, P.L., Mareschal, B., 1990. Nuclear waste management: an application of the multicriteria PROMETHEE methods. European Journal of Operational Research 44, 1–10.

Brignole, E.A., Bottini, S., Gani, R., 1986. A strategy for the design and selection of solvents for separation processes. Fluid Phase Equilibria 29, 125–132.

Brooke, A., Kendrick, D., Meeraus, A., 1988. Gams: A User's Guide. Course Technology.

Cabot, J., Teniente, E., 2007. Transformation techniques for OCL constraints. Science of Computer Programming 68, 179–195.

Catoire, F., Staicu-Casagrande, E.M., Nekkab, M., Cappello, C.D., Bartschat, K., Lahmam-Bennani, A., 2006. Investigation of the (e, 2e) single ionization of He and Ar at large energy loss close to minimum momentum transfer. Journal of Physics B: Atomic, Molecular and Optical Physics 39, 2827–2838.

Cauvin, A., 2005. Analyse, Modelisation et Amelioration de la reactivité des systèmes de décision dans les organisation industrielles.Habilitation à Diriger des Recherches. Université Paul Cezanne Aix-Marseille III.

Churi, N., Achenie, L.E.K., 1996. Novel Mathematical Programming Model for Computer Aided Molecular Design. Industrial & Engineering Chemistry Research 35, 3788–3794.

Churi, N., Achenie, L.E.K., 1997. The optimal design of refrigerant mixtures for a two-evaporator refrigeration system. Computers & Chemical Engineering 21, Supplement, S349–S354.

Cismondi, M., Brignole, E.A., 2004. Molecular Design of Solvents: An Efficient Search Algorithm for Branched Molecules. Ind. Eng. Chem. Res. 43, 784–790.

Combemale, B., 2009. Ingénierie Dirigée par les modèles (IDM) -- Etat de l'art [WWW Document]. http://hal.archives-ouvertes.fr/docs/00/37/15/65/PDF/mde-stateoftheart.pdf.

Constantinou, L., Bagherpour, K., Gani, R., Klein, J.A., Wu, D.T., 1996. Computer aided product design: problem formulations, methodology and applications. Computers & Chemical Engineering 20, 685–702.

Conte, E., 2010. Innovation in Integrated Chemical Product-process Design: Development Through a Model-based System Approach : PhD Thesis. Computer Aided Process and Product Engineering Center, Technical University of Denmark.

Conte, E., Gani, R., 2011. Chemicals-Based Formulation Design, in: Computer Aided Chemical Engineering. Elsevier, pp. 1588–1592.

Conte, E., Gani, R., Ng, K.M., 2011. Design of formulated products: A systematic methodology. AIChE Journal 57, 2431–2449.

Conte, E., Martinho, A., Matos, H.A., Gani, R., 2008. Combined Group-Contribution and Atom Connectivity Index-Based Methods for Estimation of Surface Tension and Viscosity. Ind. Eng. Chem. Res. 47, 7940–7954.

Coulin, C.R., 2008. A situational approach and intelligent tool for collaborative requirements elicitation. PhD Thesis. Université de Toulouse.

Dalkey, N., Helmer, O., 1963. An experimental application of the Delphi method to the use of experts. Management science 9, 458–467.

Duvedi, A., Achenie, L.E.K., 1997. On the design of environmentally benign refrigerant mixtures: a mathematical programming approach. Computers & Chemical Engineering 21, 915–923.

Duvedi, A.P., Achenie, L.E.K., 1996. Designing environmentally safe refrigerants using mathematical programming. Chemical Engineering Science 51, 3727–3739.

Favre, J.-M., Estublier, J., Blay-Fornarino, M., 2006. L'ingénierie dirigée par les modèles au-delà du MDA, IC2. Hermès science Lavoisier, Paris.

Fredenslund, A., Jones, R.L., Prauanitz, J.M., 1975. Group-Contribution Estimation of Activity Coefficients in Nonideal Liquid Mixtures. AIChE Journal 21, 1086–1099.

Frenkel, M., CHIRICO, R.D., DIKY, V., DONG, Q., MARSH, K.N., DYMOND, J.H., WAKEHAM, W.A., STEIN, S.E., KÖNIGSBERGER, E., GOODWIN, A.R.H., 2006. XML-based IUPAC Standard for experimental, predicted, and critically evaluated thermodynamic property data storage and capture (ThermoML). Pure Appl. Chem. 78, 541–612.

Gani, R., Brignole, E.A., 1983. Molecular design of solvents for liquid extraction based on UNIFAC. Fluid Phase Equilibria 13, 331–340.

Gani, R., Nielsen, B., Fredenslund, A., 1991. A group contribution approach to computer-aided molecular design. AIChE Journal 37, 1318–1332.

Gmehling, J., Li, J., Schiller, M., 1993. A modified UNIFAC model. 2. Present parameter matrix and results for different thermodynamic properties. Ind. Eng. Chem. Res. 32, 178–193.

Harper, P.M., Gani, R., Kolar, P., Ishikawa, T., 1999. Computer-aided molecular design with combined molecular modeling and group contribution. Fluid Phase Equilibria 158–160, 337–347.

Heintz, J., Gerbaud, V., Belaud, J.-P., 2011. Models driven conception of an Computer Aided Mixture Design tool, in: 21st European Symposium on Computer Aided Process Engineering. Elsevier, pp. 1608–1612.

Heintz, J., Gerbaud, V., Belaud, J.-P., 2012. Operational and strategic alignment in the decision process of molecule substitution, in: 22nd European Symposium on Computer Aided Process Engineering. Elsevier, pp. 382–386.

Heintz, J., Touche, I., dos Santos, M.T., Gerbaud, V., 2012. An integrated framework for product formulation by computer aided mixture design, in: 22nd European Symposium on Computer Aided Process Engineering. Elsevier, pp. 702–706.

Hellouin, L., 2002. Contribution à l'ingénierie des exigences et à la traçabilité. PhD Thesis. Université de Toulouse.

Holland, J.H., 1975. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press.

Holliday, G.L., Murray-Rust, P., Rzepa, H.S., 2006. Chemical Markup, XML, and the World Wide Web. 6. CMLReact, an XML Vocabulary for Chemical Reactions. J. Chem. Inf. Model. 46, 145–157.

Homburg, C., 1998. Hierarchical multi-objective decision making. European Journal of Operational Research 105, 155–161.

Howard, R.A., 1966. Decision analysis: Applied decision theory, in: Proceedings of the 4th International Conference on Operational Research. pp. 55–77.

Huber, G.P., 1980. Managerial decision making. Scott, Foresman.

Hukkerikar, A.S., Sarup, B., Ten Kate, A., Abildskov, J., Sin, G., Gani, R., 2012. Group-contribution+ (GC+) based estimation of properties of pure components: Improved property estimation and uncertainty analysis. Fluid Phase Equilibria 321, 25–43.

IBM, 2011. Rational Software White Paper: Rational Unified Process Best Practices for Software Development Teams.

IEEE, 1990. IEEE Std 610.12-1990 Standard Glossary of Software Engineering Terminology.

IEEE, 2000. Recommended Practice for Architectural Description of Software-Intensive Systems, IEEE-std-1471-2000.

ISO TC 184/SC 5, 2000. Industrial Automation Systems - Requirements for enterprise-reference architectures and methodologies (patent No. ISO 15704:2000).

Izza, S., 2006. Intégration des Systèmes d'Information Industriels-Une approche flexible basée sur les services sémantiques. PhD Thesis.Ecole Nationale Supérieure des Mines de Saint-Etienne.

Joback, K.G., Reid, R.C., 1987. ESTIMATION OF PURE-COMPONENT PROPERTIES FROM GROUP-CONTRIBUTIONS. Chemical Engineering Communications 57, 233–243.

Joback, K.G., Stephanopoulos, G., 1995. Searching Spaces of Discrete Solutions: The Design of Molecules Possessing Desired Physical Properties, in: Intelligent Systems in Process Engineering Part I: Paradigms from Product and Process Design. Academic Press, pp. 257–311.

Keeney, R.L., Raiffa, H., 1993. Decisions with Multiple Objectives: Preferences and Value Trade-Offs. Cambridge University Press.

Klein, J.A., Wu, D.T., Gani, R., 1992. Computer aided mixture design with specified property constraints. Computers & Chemical Engineering 16, Supplement 1, S229–S236.

Konaté, J., 2010. Approche système pour la conception d'une méthodologie pour l'élicitation collaborative des exigences. PhD Thesis. Université de Toulouse.

Korichi, M., 2010. Conception des produits- Procédés Assistée par Ordinateur: Formulation et Application aux Arômes. PhD Thesis. Université d'Oran.

Korichi, M., Gerbaud, V., Floquet, P., Meniai, A.-H., Nacef, S., Joulia, X., 2008. Computer aided aroma design I-Molecular knowledge framework. Chemical Engineering and Processing: Process Intensification 47, 1902–1911.

Kroll, P., Kruchten, P., 2003. The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP, 1st ed. Addison Wesley.

LAAS-CNRS, 2003. Rapport final de l'Action Spécifique n°35 - AS PRODLOG "Production et logistique dans l'entreprise étendue: modèles et outils collaboratifs."

Le Moigne, J.-L., 1994. La théorie du système général théorie de la modélisation, 4e édition mise à jour. ed, Systèmes-décisions. Presses universitaires de France, Paris.

Liaw, H.-J., Gerbaud, V., Li, Y.-H., 2011. Prediction of miscible mixtures flash-point from UNIFAC group contribution methods. Fluid Phase Equilibria 300, 70–82.

Liebman, J., Lasdon, L., Schrage, L., Waren, A., 1986. Modeling and Optimization With Gino. Brooks/Cole.

Lin, B., Chavali, S., Camarda, K., Miller, D.C., 2005. Computer-aided molecular design using Tabu search. Computers & Chemical Engineering 29, 337–347.

Linstone, H.A., Turoff, M., 1975. The Delphi method: techniques and applications. Addison-Wesley Pub. Co., Advanced Book Program.

Maranas, C.D., 1996. Optimal Computer-Aided Molecular Design: A Polymer Design Case Study. Ind. Eng. Chem. Res. 35, 3403–3414.

Maranas, C.D., 1997. Optimization accounting for property prediction uncertainty in polymer design. Comput. Chem. Eng. 21, S1019–S1024.

Marcoulaki, E.C., Kokossis, A.C., 1998. Molecular design synthesis using stochastic optimisation as a tool for scoping and screening. Computers & Chemical Engineering 22, Supplement 1, S11–S18.

Marcoulaki, E.C., Kokossis, A.C., 2000a. On the development of novel chemicals using a systematic synthesis approach. Part I. Optimisation framework. Chemical Engineering Science 55, 2529–2546.

Marcoulaki, E.C., Kokossis, A.C., 2000b. On the development of novel chemicals using a systematic optimisation approach. Part II. Solvent design. Chemical Engineering Science 55, 2547–2561.

Marquardt, W., Wedel, L., Bayer, B., 2000. Perspectives on lifecycle process modeling, in: Proceedings on Foundations of Computer-Aided Process Design, AIChE Symposium Series 323.

Marrero, J., Gani, R., 2001. Group-contribution based estimation of pure component properties. Fluid Phase Equilibria 183–184, 183–208.

Marrero, J., Gani, R., 2002. Group-Contribution-Based Estimation of Octanol/Water Partition Coefficient and Aqueous Solubility. Ind. Eng. Chem. Res. 41, 6623–6633.

Martin, R.A., Robertson, E.L., Springer, J.A., 2004. Architectural principles for enterprise frameworks, in: CAiSE Workshops Knowledge and Model Driven Information Systems Engineering for Networked Organizations.

Martin, T.M., Young, D.M., 2001. Prediction of the Acute Toxicity (96-h LC50) of Organic Compounds to the Fathead Minnow (Pimephales promelas) Using a Group Contribution Method. Chem. Res. Toxicol. 14, 1378–1385.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics 21, 1087.

Mintzberg, H., Raisinghani, D., Theoret, A., 1976. The structure of“ unstructured” decision processes. Administrative science quarterly 246–275.

Nuseibeh, B., Easterbrook, S., 2000. Requirements engineering: a roadmap, in: Proceedings of the Conference on the Future of Software Engineering. pp. 35–46.

Odele, O., Macchietto, S., 1993. Computer aided molecular design: a novel method for optimal solvent selection. Fluid Phase Equilibria 82, 47–54.

OMG, 2006. Object Constraint Language OMG Available Specification Version 2.0 [WWW Document]. http://www.omg.org/spec/OCL/2.3.1/PDF/.

OMG, 2008. Semantics of Business Vocabulary and Business Rules (SBVR), v1.0 ,OMG Available Specification [WWW Document]. http://www.omg.org/spec/SBVR/1.0/PDF/.

Ourique, J.E., Silva Telles, A., 1998. Computer-aided molecular design with simulated annealing and molecular graphs. Computers & Chemical Engineering 22, Supplement 1, S615–S618.

Patkar, P.R., Venkatasubramanian, V., 2002. Chapter 5: Genetic algorithms based CAMD, in: Computer Aided Molecular Design: Theory and Practice. Elsevier, pp. 95–128.

Perez, J.M., Ruiz, F., Piattini, M., 2008. MDE for BPM: A Systematic Review, in: Filipe, J., Shishkov, B., Helfert, M. (Eds.), Software and Data Technologies, Communications in Computer and Information Science. Springer Berlin Heidelberg, pp. 127–135.

Pourcel, C., Gourc, D., 2005. Modélisation d’entreprise par les processus activités, organisation & applications. Cépaduès-éd, Toulouse.

Pretel, E.J., López, P.A., Bottini, S.B., Brignole, E.A., 1994. Computer-aided molecular design of solvents for separation processes. AIChE Journal 40, 1349–1360.

Raman, V.S., Maranas, C.D., 1998. Optimization in product design with properties correlated with topological indices. Computers & Chemical Engineering 22, 747–763.

Reid, R.C., Prausnitz, J.M., Poling, B.E., 1987. The Properties of Gases and Liquids, 4th Revised

ed. McGraw-Hill Inc.,US.

Riedel, L., 1954. Eine neue universelle Dampfdruckformel Untersuchungen über eine Erweiterung des Theorems der übereinstimmenden Zustände. Teil I. Chemie Ingenieur Technik 26, 83–89.

Ross, R.G., 2003. Principles of the Business Rule Approach, 1st ed. Addison Wesley.

Roy, B., 1968. Classement et choix en présence de points de vue multiples. Revue française d'automatique, f'informatique et de recherche opérationnelle 2, 57–75.

Saaty, T.L., 2001. The Analytic Network Process: Decision Making With Dependence and Feedback, 2nd ed. Rws Pubns.

Saaty, T.L., Vargas, L., 1994. Fundamentals of Decision Making and Priority Theory With the Analytic Hierarchy Process. Rws Pubns.

Simon, H.A., 1960. The new science of management decision, The ford distinguished lectures. Harper and Row, New York and Evanston.

Sinha, M., Achenie, L.E.., Ostrovsky, G.M., 1999. Environmentally benign solvent design by global optimization. Computers & Chemical Engineering 23, 1381–1394.

Sinha, M., Achenie, L.E.K., 2002. Chapter 11: CAMD in solvent mixture design, in: Computer Aided Molecular Design: Theory and Practice. Elsevier, pp. 261–287.

Smith, E.G., Wiswesser, W.J., Addelston, A., 1968. The Wiswesser line-formula chemical notation. McGraw-Hill.

Smith, J.M., Ness, H.C.V., Abbott, M., Ness, H.V., 2000. Introduction to Chemical Engineering Thermodynamics, 6th ed. McGraw-Hill Science/Engineering/Math.

Solomakhin, D., 2011. Logical Formalization of Semantic Business Vocabulary and Rules. Master Thesis. Faculty of Informatics, Vienna University of Technology.

Song, J., Song, H.-H., 2008. Computer-Aided Molecular Design of Environmentally Friendly Solvents for Separation Processes. Chemical Engineering & Technology 31, 177–187.

Tavana, M., Pirdashti, M., Kennedy, D.T., Belaud, J.-P., Behzadian, M., 2012. A hybrid Delphi-SWOT paradigm for oil and gas pipeline strategic planning in Caspian Sea basin. Energy Policy 40, 345–360.

Thery, R., Zarate, P., 2009. Energy planning: a multi-level and multicriteria decision making structure proposal. Central European Journal of Operations Research 17, 265–274.

Touzi, J., 2008. Aide à la conception de Système d'Information Collaboratif , support de l'interopérabilité des entreprises. PhD Thesis, Université de Toulouse.

Touzi, J., Benaben, F., Pingaud, H., Lorré, J.P., 2009. A model-driven approach for collaborative service-oriented architecture design. International Journal of Production Economics 121, 5–20.

Ulmer, J.S., 2011. Approche générique pour le modélisation et l'implémentation des processus. PhD Thesis. Université de Toulouse.

Ulmer, J.S., Belaud, J.. P., Le Lann, J.M., 2011. Towards a pivotalbased approach for business process alignment. International Journal of Computer Integrated Manufacturing.

Vaidyanathan, R., El-Halwagi, M., 1996. Computer-Aided Synthesis of Polymers and Blends with Target Properties. Ind. Eng. Chem. Res. 35, 627–634.

Vaidyanathan, R., Gowayed, Y., El-Halwagi, M., 1998. Computer-aided design of fiber reinforced polymer composite products. Computers & Chemical Engineering 22, 801–808.

Vaidyaraman, S., Maranas, C.D., 1999. Optimal synthesis of refrigeration cycles and selection of refrigerants. AICHE J. 45, 997–1017.

Vallin, P., Vanderpooten, D., 2000. Aide à la décision une approche par les cas gestion, mathématiques, informatique. Ellipses, Paris.

van Dyk, B., Nieuwoudt, I., 2000. Design of Solvents for Extractive Distillation. Ind. Eng. Chem. Res. 39, 1423–1429.

Veith, G.D., Konasewich, D.E., 1975. Structure-activity Correlations in Studies of Toxicity and Bioconcentration with Aquatic Organisms. Great Lakes research advisory Board, Windsor/Ontario.

Vernadat, F., 1999. Techniques de modélisation en entreprise applications aux processus opérationnels, Collection Gestion. Économica, Paris.

Wan-Kadir, W.M.N., Loucopoulos, P., 2004. Relating evolving business rules to software design. Journal of Systems Architecture 50, 367–382.

Weininger, D., 1988. Smiles, a Chemical Language and Information-System .1. Introduction to Methodology and Encoding Rules. J. Chem. Inf. Comput. Sci. 28, 31–36.

Weininger, D., Weininger, A., Weininger, J., 1989. Smiles .2. Algorithm for Generation of Unique Smiles Notation. J. Chem. Inf. Comput. Sci. 29, 97–101.

Weis, D.C., Visco, D.P., 2010. Computer-aided molecular design using the Signature molecular descriptor: Application to solvent selection. Computers & Chemical Engineering 34, 1018–1029.

White, S.A., 2004. Introduction to BPMN [WWW Document]. URL http://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf

Wiegers, K.E., 2009. Software Requirements. O'Reilly Media, Inc.

CHAPTER 10

Appendixes

The appendixes are the following:

10.1	Basic functional group coding	.....	194
10.2	List of calculation models	.....	199
10.3	UML Diagrams	.....	203
10.4	Graph Construction method	.....	205
10.5	Composition handling	.....	216
10.6	IBSS XML input file	.....	219
10.7	IBSS output file extract	.....	228
10.8	Delphi questionnaire	.....	247
10.9	Acronyms	.....	255

10.1 BASIC FUNCTIONAL GROUP CODING

The basic functional group coding used in our work is based on the one proposed by Korichi et al. (2008) is the following.

A functional group is coded as EG = $P_1P_2P_3P_4$

With P1 the atomic number, preceded with a 1 (106 for C, 107 for N, 108 for O, 117 for Cl,…)

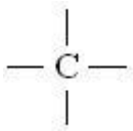
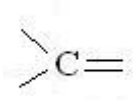
P2 the highest bond order on the atom (ex. 1 for « -C- », 2 for « -N= », 3 for « -C≡ » and a special case 4 for =C=

P3 the bond type (0 for bond with C or with the same atom. e.g. C-C-C; N-N; 1 for bond with at least one non-similar atom (C-O, C=O, C≡N, N-O); 2 for atom in a non aromatic cycle e.g. pyridine, 3 in an aromatic ring e.g. benzene; 4 for an atom shared by two aromatic rings e.g. naphtalene; 5 for an atom shared by two cycles, one of them aromatic e.g. indane, 6 for other aromatic cases e.g. biphenyl; 7 for an atom shared by two non-aromatic rings.

P4 is the number of implicit hydrogens

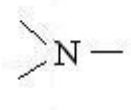
The list of the available basic groups and their encoding are:

Carbon:

Atom C	EG	H	Comments	
	10610H	0 – 3	– CH3 – CH2 – CH – C	106103 106102 106101 106100
	10611H	0 – 3	X – CH3 X – CH2 X – CH X – C	106113 106112 106111 106110
	10612H	0 – 2	ring non arom-CH2 ring non arom-CH ring non arom-C	106122 106121 106120
	10616H	0 – 1	ring-sp-CH ring-sp-C	106161 106160
	10620H	0 – 2	= CH2 = CH = C	106202 106201 106200

	10621H	0 – 2	X = CH2 X = CH X = C	106212 106211 106210
	10622H	0 – 1	ring non-arom-CH ring non-arom-C	106221 106220
	10623H	0 – 1	ring arom-CH ring arom-C	106231 106230
	106240	0	Common atom between two aromatic cycles	106240
	106250	0	Common atom between two – 1 aromatic and 1 non aromatic	106250
– C ≡	10630H	0 – 1	≡ CH ≡ C	106301 106300
	10631H	0 – 1	X ≡ CH X ≡ C	106311 106310
	106320	0	ring non-arom ≡ C	106320
= C =	106400	0	connection to C only	106400
	106410	0	one non-C connection	106410
	106420	0	non-aromatic cycle	106420

Nitrogen:

Atom N	EG	H	Comments	
	10710H	0 – 2	– NH2 – NH – N	107102 107101 107100
	10711H	0 – 2	X – NH2 X – NH X – N	107112 107111 107110
	10712H	0 – 1	ring non-arom-NH ring non-arom-N	107121 107120

– N =	10720H	0 – 1	= NH = N	107201 107200
	10721H	0 – 1	X = NH X = N	107211 107210
	107220	0	non-aromatic cycle	107220
	107230	0	aromatic cycle	107230
N ≡	107300	0	connection to C only	107300
	107310	0	one non-C connection	107310

Oxygen:

Atom O	EG	H	Comments	
– O –	10810H	0 – 1	– OH – O	108101 108100
	10811H	0 – 1	X– OH X– O	108111 108110
	108120	0	non-aromatic cycle	108120
	108130	0	aromatic cycle	108130
O =	108200	0	connection to C only	108200
	108210	0	one non-C connection	108210
	108220	0	non-aromatic cycle	108220

Halogens:

	EG	H	Commentaires	
– Cl	117100	0	connection to C only	117100
	117110	0	one non-C connection	117110
– F	109100	0	connection to C only	109100
	109110	0	one non-C connection	109110

– Br	135100	0	connection to C only	135100
	135110	0	one non-C connection	135110
– I	153100	0	connection to C only	153100
	153110	0	one non-C connection	153110

Phosphor:

	EG	valency	Comments	
P	11510H	valency 3	C – PH2	115102
			C – PH	115101
			C – P	115100
			X – PH2	115112
			X – PH	115111
			X – P	115110
P	1152*0		2 connections: 1 simple + 1 double	115200 115210
HP	1152*1		1 double connection	115201 115211
PH	1151*1	valency 5	4 simple connections	115101 115111
P	1152*0		4 connections: 3 simple + 1 double	115200 115210
P	1154*0		3 connections: 1 simple + 2 double	115400 115410
PH	1154*1		2 connections: 2 doubles	115401 115411

*: 0 if all connections are with C and/or P, 1 if at least one connection is with on atom different than C or P.

Sulphur:

	EG	valency	Comments

HS	1161*1	valency 2	1 simple connection	116101 116111
S	1161*0		2 simple connections	116100 116110
SH	1161*1	valency 4	3 simple connections	116101 116111
S	1161*0		4 simple connections	116100 116110
SH	1162*1		2 connections: 1 simple + 1 double	116201 116211
S	1162*0		3 connections: 2 simple + 1 double	116200 116210
S	1164*0		2 double connections	116400 116410
SH	1162*1	valency 6	4 connections: 3 simples+ 1 double	116201 116211
S	1164*0		4 connections: 2 simple+ 2 double	116400 116410
SH	1164*1		3 connections: 1 simple + 2 double	116401 116411
S			3 double connections	

*: 0 if all connections are with C and/or S, 1 if at least one connection is with on atom different than C or S.

10.2 LIST OF CALCULATION MODELS

- 101_MolecWeight_mix
  - 101_MolecWeight_pure_MT2010
- 102_MeltingPoint_mix
  - 102_MeltingPoint_pure_MG2001
  - 102_MeltingPoint_pure_CG1994
  - 102_MeltingPoint_pure_JR1987
  - 102_MeltingPoint_pure_JYY2004
- 103_BoilingPoint_mix
  - 103_BoilingPoint_pure_MG2001
  - 103_BoilingPoint_pure_CG1994
  - 103_BoilingPoint_pure_JR1987
- 104_CriticalTemp_mix
  - 104_CriticalTemp_pure_MG2001
  - 104_CriticalTemp_pure_CG1994
- 105_CriticalPres_mix
  - 105_CriticalPres_pure_MG2001
  - 105_CriticalPres_pure_CG1994
  - 105_CriticalPres_pure_JR1987
- 106_CriticalVol_mix
  - 106_CriticalVol_pure_MG2001
  - 106_CriticalVol_pure_CG1994
  - 106_CriticalVol_pure_JR1987
- 107_GibbsEnergy298K_mix
  - 107_GibbsEnergy298K_pure_MG2001
  - 107_GibbsEnergy298K_pure_CG1994
  - 107_GibbsEnergy298K_pure_JR1987
- 108_EnthalpyFormation298K_mix
  - 108_EnthalpyFormation298K_pure_MG2001
  - 108_EnthalpyFormation298K_pure_CG1994

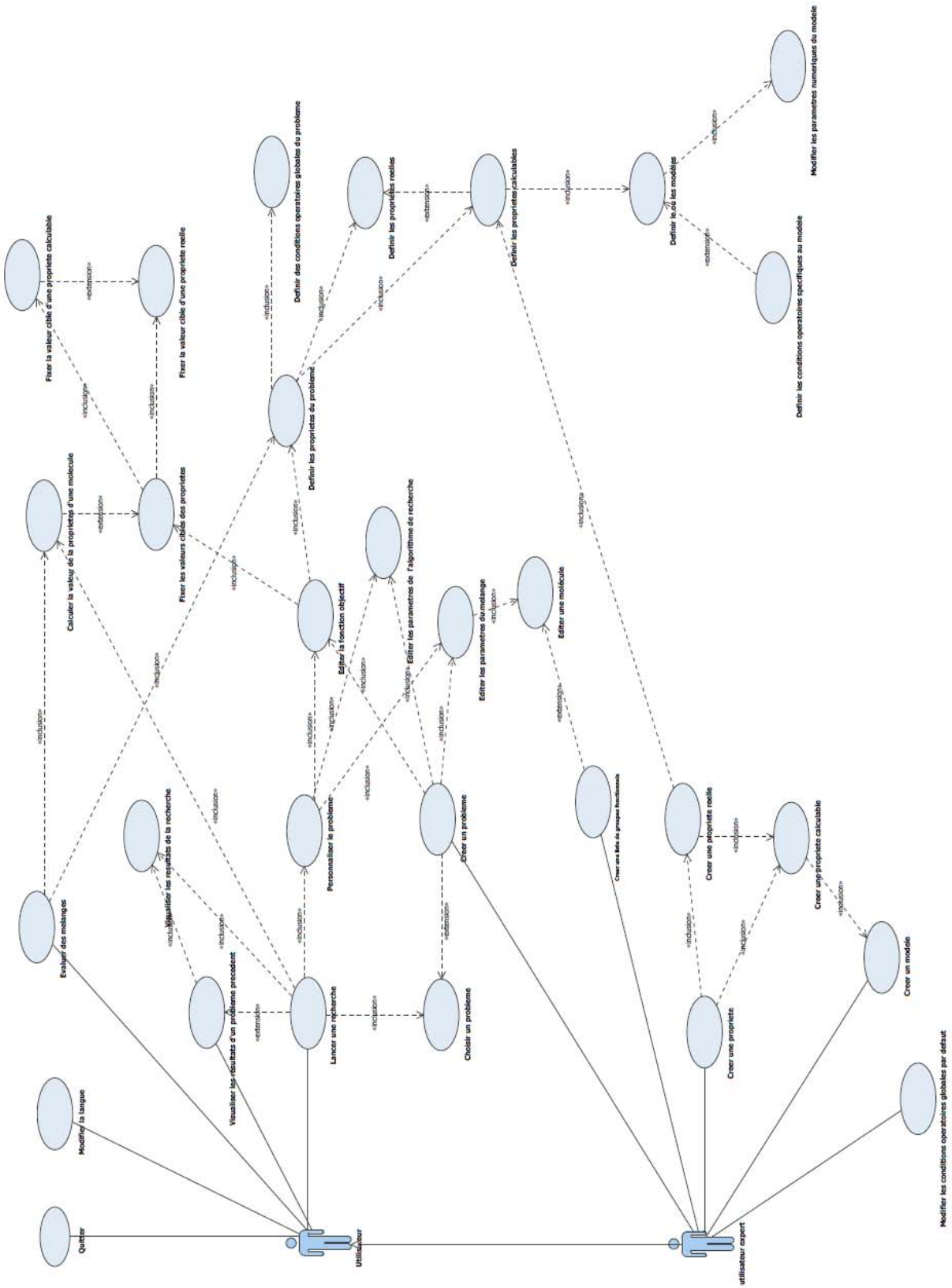
    - 108_EnthalpyFormation298K_pure_JR1987
- 109_EnthalpyVaporization298K_mix
    - 109_EnthalpyVaporization298K_pure_MG2001
    - 109_EnthalpyVaporization298K_pure_CG1994
    - 109_EnthalpyVaporization298K_pure_JR1987
- 110_EnthalpyFusion_mix
    - 110_EnthalpyFusion_pure_MG2001
    - 110_EnthalpyFusion_pure_JR1987
- 111_VaporPressure_mix,Temp_op,25,°C
    - 111_VaporPressure_pure_Riedel1954,Temp_op,25,°C
- 112_FlashPoint_mix
    - 112_FlashPoint_pure_CPN2006
- 113_HansenD_mix
    - 113_HansenD_pure_MG2001
    - 113_HansenD_pure_JR1987
- 113_HansenD_mix_MB2010
    - 113_HansenD_pure_MB2010
- 114_HansenP_mix
    - 114_HansenP_pure_MG2001
    - 114_HansenP_pure_JR198
- 114_HansenP_mix_MB20107
    - 114_HansenP_pure_MB2010
- 115_HansenH_mix
    - 115_HansenH_pure_MG2001
    - 115_HansenH_pure_JR1987
- 115_HansenH_mix_MB2010
    - 115_HansenH_pure_MB2010
- 116_MolecVolume_mix
    - 116_MolecVolume_pure_MB2010
- 117_HD_mix
    - 117_HD_pure_MB2010

- 118_HA_mix
    - 118_HA_pure_MB2010
- 119_Density_mix
    - 19_Density_pure_YG1993
- 119_Density_mix_MB2010
    - 119_Density_pure_MB2010
- 120_AcentricFactor_mix
    - 120_AcentricFactor_pure_CGOC1995
- 121_Viscosity_mix,Temp_op,25,°C
    - 121_Viscosity_pure_CMMG2008,Temp_op,25,°C
    - 121_Viscosity_mix_TejaRice81,Temp_op,25,°C
- 122_SurfaceTension_mix,Temp_op,25,°C
    - 122_SurfaceTension_pure_CMMG2008,Temp_op,25,°C
    - 122_SurfaceTension_mix_TKO55,Temp_op,25,°C
- 123_HSPDistance_mix,HSPd_target,0,HSPp_target,0,HSPh_target,0
    - 123_HSPDistance_pure_MB2010,HSPd_target,0,HSPp_target,0,HSPh_target,0
    - 123_HSPDistance_mix_MB2010,HSPd_target,0,HSPp_target,0,HSPh_target,0
- 124_EnvWaste_mix
    - 124_EnvWaste_pure_Weis2009
- 125_EnvImpact_mix
    - 125_EnvImpact_pure_Weis2009
- 126_Health_mix
    - 126_Health_pure_Weis2009
- 127_Safety_mix
    - 127_Safety_pure_Weis2009
- 128_LCA_mix
    - 128_LCA_pure_Weis2009
- 129_Kow_mix
    - 129_Kow_pure_KowHG
    - 129_Kow_pure_MG2002

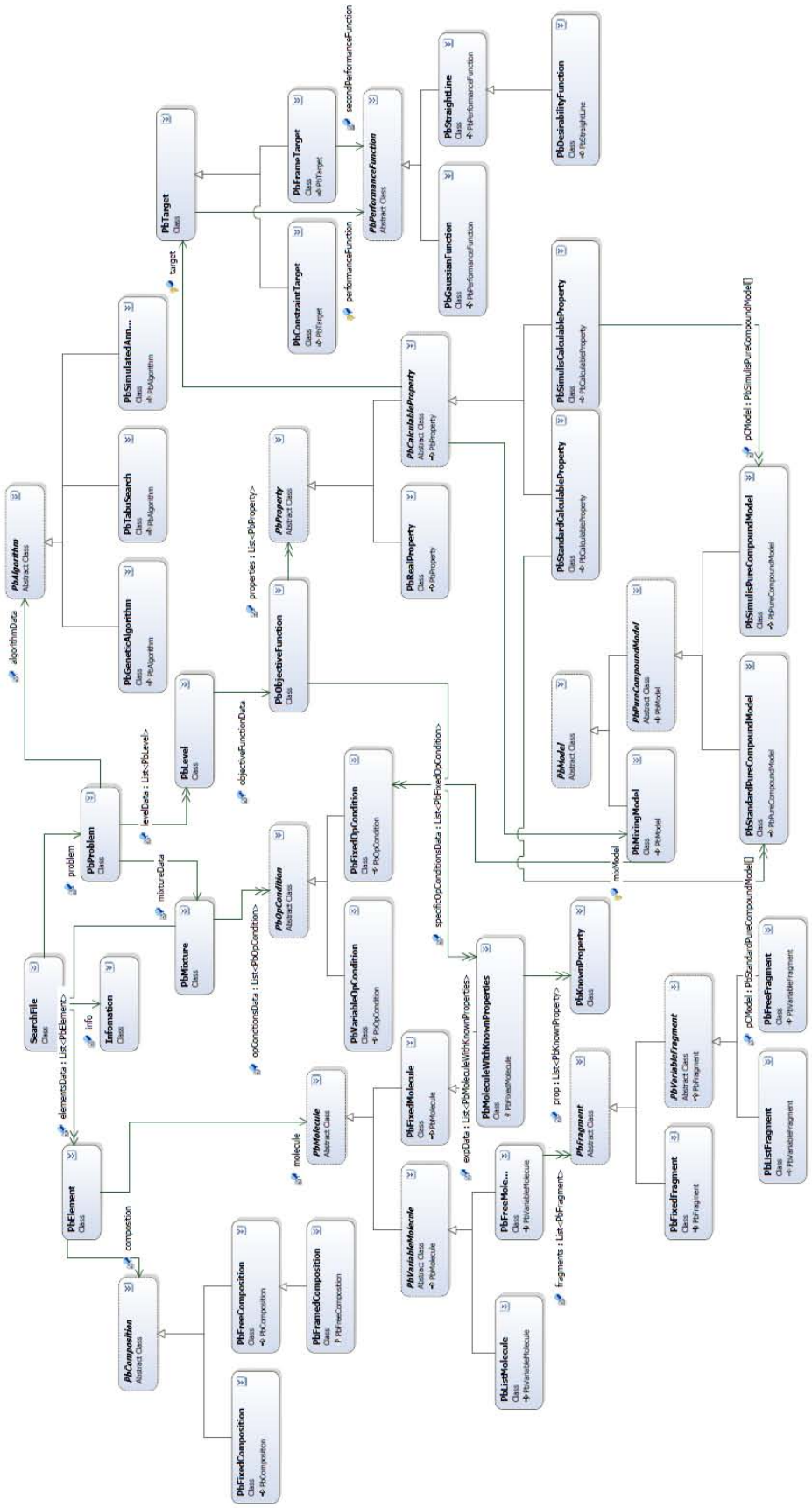
- 130_MeltingEntropy_mix
    - 130_MeltingEntropy_pure_JYY2004
- 131_AqueosSolubility_mix
    - 131_AqueosSolubility_pure_MG2002
- 132_RED_mix,HSPradius_target,1,HSPd_target,0,HSPp_target,0,HSPh_target,0
    - 132_RED_pure_MB2010,HSPradius_target,1,HSPd_target,0,HSPp_target,0,HSPh_target,0
- 132_RED_mix_MB2010,HSPradius_target,1,HSPd_target,0,HSPp_target,0,HSPh_target,0
    - 132_RED_pure_MB2010,HSPradius_target,1,HSPd_target,0,HSPp_target,0,HSPh_target,0
- 133_LC50_mix
    - 33_LC50_pure_MY2001
    - 133_LC50_pure_Konemann1981
- 134_BCF_mix
    - 134_BCF_pure_VK1975

10.3 UML DIAGRAMS

10.3.1 Use Case Diagram



10.3.2 Problem Package Class Diagram



10.4 GRAPH CONSTRUCTION METHOD

This section describes, through an example, the method implemented in the GraphConstructor class.

10.4.1 Initialization

The method input parameters are:

- kmax: the maximum number of the functional groups in the graph
- kmin: the minimum number of the functional groups in the graph
- probAromatic: probability for a cycle to be aromatic
- listM: the list of the possible values of the variable "m"is equal to 1 –the number of cycles in the graph
- listExtConnections: the list of the external connection with specifications on the type
- listOfSetOfFgroup: list of the functional groups that can be used, ordered in sets considering the number and the type of their attachments

The method output parameters are:

- fragmentGraph: graph of the fragment
- setGroupReference: vector containing the reference of the set of the functional groups in listOfSetOfFGroup
- connection: vector containing the information of the location of the external connections in the graph
- cycleList: list of the cycles in the graph and the elements that form them
- cycleListAromatic: list specifying the aromaticity of the cycles in cycleList

The input data are transmitted by a freefragment object and correspond to the user constraints. In order to guarantee the diversity of the generated graphs, several choices are randomly made:

- Choice of k, the number of groups in the fragment between kmin and kmax
- Choice of m, which is equal to 1 –the number of cycles, among the values in listM

The graph could be constructed with these two pieces of information but the chances to reach dead ends are high and the diversity of the structure generated may be poor. We thus choose to use group vectors. A group vector VG can be represented the following way:

$$VG = \{N_1 \quad N_2 \quad \dots \quad N_n\}$$

Where:

- N_1 is the number of groups in the fragment that have one connection
- N_2 is the number of groups in the fragment that have two connections
- N_n is the number of groups in the fragment that have “n” connections

It must respect the following chemical feasibility rules coming from the octet rule:

- $\sum_{j=1}^{n} N_j = k$
- $\sum_{j=1}^{n} N_j * (2 - j) = 2m - extConnections$

Where:

- N_j is the number of group in the fragment that have “j” connections
- k is the number of group in the fragment
- m is a number is equal to 1 –the number of cycles
- $extConnections$ is the number of external connections of the fragment

A group vector is randomly chosen among all the possibilities. This allows constraining the structures in order to avoid dead ends. The fact that each possibility has the same probability to be chosen ensures a higher diversity of the generated structures.

The following example explains how the choices are made.

Relevant input data:
- 1 external connection
- kmin = 1
- kmax = 10
- listM={0; 1}
- the maximum number of attachments of the groups is 4

Choice of k	$_{kmin}\{1; 2; 3; 4 ; 5; 6; 7; 8; 9; 10\}_{kmax}$
Choice of m	$\{0; 1\}$
Calculation of all the possible VG and choice of one	N_1 N_2 N_3 N_4 1 5 0 0 2 3 1 0 3 1 2 0 3 2 0 1 4 0 1 1

The number of functional groups in the graph, k, is randomly chosen between kmin and kmax, here k=6. The value of the variable m is then chosen in the values of listM, here m=1 which means that the structure will not have cycles. Then all the possible group vectors are generated. In this example, there are 5 possibilities. A group vector is randomly chosen and it will be used as basis for the graph construction. Here the graph will contain three groups with one connection, two groups with two connections and one group with four connections.

10.4.2 Graph generation

Once the group vector is chosen the groups are chosen one by one, respecting the group vector and the number of cycles. If a choice leads to a dead-end, another path is considered.

During the generation, the external connections are considered as terminal functional groups (groups with a single attachment) and are temporally written in the graph. In order to have a graph easy to read, the graph always starts with a connection. If the graph has no external connection, it starts with a cycle or a terminal group. When the graph is complete, all the parameters are updated to extract the connections from the graph.

Let us continue the example. Our objective here is not to be exhaustive but to show the main mechanisms of a random graph construction. This example is very simple (there is no cycle to be constructed) but it gives a good overview of how the elements are inserted, which tests are made and how dead ends are avoided. More information about how cycles are inserted to the graph is given in 4.6.4

We have the following relevant input data:

Number of cycles: 0

1 external connection of type 1

Group Vector: (3 2 0 1)

List of the Sets of groups

reference of the group set			
1	$CH3-$	$OH-$	$NH2-$
2	$-CH2-$	$-O-$	$-NH-$
3	$>CH-$	$>N-$	
4	$>C<$		
5	$CH2=$	$O=$	$NH=$
6	$-CH=$	$-N=$	
7	$>C=$		
8	$CH\equiv$		
9	$-C\equiv$		

The graph to be constructed must have an external connection of type 1 (single bound). It represents a molecular fragment. It must be conformed to the group vector and thus contains exactly 3 functional groups with 1 connection, 2 with 2 connections, 0 with 3 connections and 1 with 4 connections. The graph must contain only the functional groups specified in the list.

The list is arranged by sets of groups with the same connections. Working with set allows accessing quickly to the information of the connectivity of the group. Choosing first the set then the group allows testing early if the construction is in a dead end or not. The information of the set of the different functional groups of the fragment is precious for the modification operators; it is thus saved in the vectorial variable “setGroupReference”.

- Choice of the first element

  The first element to be inserted to the graph is the external connection in order to facilitate the reading of the molecular graph (since it is the aggregation of all the fragment graphs).

1 external connection of type 1

Group Vector: $(3 \quad 2 \quad 0 \quad 1)$

fragmentgraph setGroupReference

$$\begin{pmatrix} Ext & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

R—

- Choice of the second element

  The next element must be conformed to the group vector and have at least one connection of type 1. With this information, the sets of groups are screened and only the suitable sets are kept. One of these sets is randomly chosen and some tests (on the valency, on the possibility to construct cycle if needed, etc.) are run. Here, the addition of a group of the set number 1, which contains terminal groups, will close the fragment (valency=0) while more functional groups need to be added. Thus it is not possible to add a group of this set. It will be deleted from the list of the possible sets of groups and another set will be randomly chosen. The tests are run with this new set and if they are passed then a functional group of this set will be randomly chosen and added in the fragment graph. The group vector is then updated in order to only present the information of the groups that still have to be inserted.

Group Vector: $(3 \quad 2 \quad 0 \quad 1)$

Connection of type 1

Choice of a set of group

- List of possible groups

  $1 \quad 2 \quad 4 \quad 6 \quad 9$

  Random choice indexed on the number of the groups in the set

  Test NOK => valence =0 and graph not complete

- List of possible groups

  ~~1~~ $2 \quad 4 \quad 6 \quad 9$

  Test OK

Choice of a functional group =>-CH2-

$$\begin{pmatrix} Ext & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & CH2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

R—CH_2—

- Choice of the third element

  The choice of the next element is then made. It is always connected to the last inserted element that has free connections. Here it will be connected to the second element via a single bond. It can be noted that the group vector has been updated.

  The list of possible sets of groups is established. A set is chosen, tests are run and a functional group is inserted.

Group Vector: $(3 \quad 1 \quad 0 \quad 1)$

Connection of type 1

Choice of a set of group

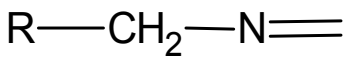
- List of possible groups

  $1 \quad 2 \quad 4 \quad 6 \quad 9$

  → Test OK

Choice of a functional group =>-N=

$$\begin{pmatrix} Ext & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & CH2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & N & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 6 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



- Choice of the fourth element

  The choice of the next element is then made. Here it will be connected to the third element via a double bond.

  The list of possible sets of groups contains only one set of groups and this set fail to the tests. This means that there are no possible set of groups to complete the graph. We call this situation a dead end. The previous choices need to be reconsidered.

Group Vector: $(3 \quad 0 \quad 0 \quad 1)$

Connection of type 2

Choice of a set of group

- List of possible groups

  5

  → Test NOK => valence =0 and graph not complete

Dead end

- Reconsideration of the third element

  As we just have noticed, the choice of the set number 6 here leads to a dead end. This set is thus removed from the list of the possible sets, and another set is chosen.

Group Vector: $(3 \quad 1 \quad 0 \quad 1)$

Connection of type 1

Choice of a set of group

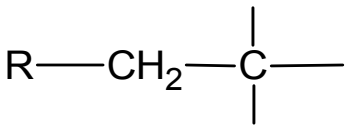
- List of possible groups

$1 \quad 2 \quad 4 \quad \cancel{6} \quad 9$

Test OK

Choice of a functional group =>>C<

$$\begin{pmatrix} Ext & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & CH2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & C & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 4 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



- Choice of the fourth element

The choice of the next element is then made. Here it will be connected to the third element via a single bond. The third element has several free connections but all of the same type. If it had free connections of different types, then a random choice would have been made in order to decide to which bound the next element will be connected with. The standard procedure for choosing the inserted functional group is followed.

Group Vector: $(3 \quad 1 \quad 0 \quad 0)$

Connection of type 1

Choice of a set of group

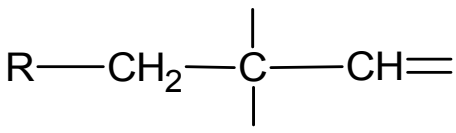
- List of possible groups

$1 \quad 2 \quad 6 \quad 9$

Test OK

Choice of a functional group =>-CH=

$$\begin{pmatrix} Ext & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & CH2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & C & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & CH & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 4 \\ 6 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$



- Choice of the fifth element

  The choice of the next element is then made. Here it will be connected to the fourth element via a double bond. The standard procedure for choosing the inserted functional group is followed.

Group Vector: $(3 \quad 0 \quad 0 \quad 0)$

Connection of type 2

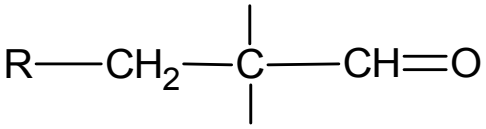
Choice of a set of group

- List of possible groups

5 → Test OK

Choice of a functional group =>O=

$$\begin{pmatrix} Ext & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & CH2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & C & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & CH & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & O & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 4 \\ 6 \\ 5 \\ 0 \\ 0 \end{pmatrix}$$



- Choice of the sixth element

  The choice of the next element is then made. As the last element inserted has no free connection, the last element inserted that still has free connections is considered. Thus, here the sixth element will be connected to the third element via a single bond. The standard procedure for choosing the inserted functional group is followed.

Group Vector: $(2 \quad 0 \quad 0 \quad 0)$

Connection of type 1

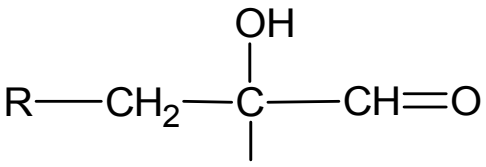
Choice of a set of group

- List of possible groups

1 → Test OK

Choice of a functional group =>OH-

$$\begin{pmatrix} Ext & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & CH2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & C & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & CH & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & O & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & OH & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 4 \\ 6 \\ 5 \\ 1 \\ 0 \end{pmatrix}$$



- Choice of the seventh element

  The choice of the next element is then made. As previously, the last element inserted has no free connection and the last element inserted that still has free connections is considered. Thus, here the seventh element will be connected to the third element via a single bond. The standard procedure for choosing the inserted functional group is followed. After this insertion, the graph is complete.

Group Vector: $(1 \quad 0 \quad 0 \quad 0)$

Connection of type 1

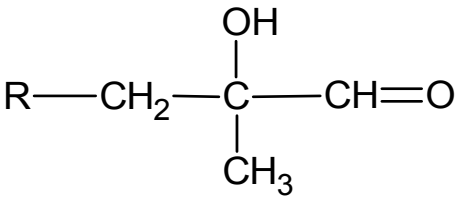
Choice of a set of group

- List of possible groups

1 → Test OK

Choice of a functional group =>CH3-

$$\begin{pmatrix} Ext & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & CH2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & C & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & CH & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & O & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & OH & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & CH3 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 4 \\ 6 \\ 5 \\ 1 \\ 1 \end{pmatrix}$$



10.4.3 Update of the parameters

When the graph is complete a last operation must be made. It consists in removing the external connections from the graph to put them in the vector “connection”.

$$\begin{pmatrix} Ext & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & CH2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & C & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & CH & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & O & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & OH & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & CH3 \end{pmatrix} \begin{pmatrix} -1 \\ 2 \\ 4 \\ 6 \\ 5 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} CH2 & 1 & 0 & 0 & 0 & 0 \\ 1 & C & 1 & 0 & 1 & 1 \\ 0 & 1 & CH & 2 & 0 & 0 \\ 0 & 0 & 2 & O & 0 & 0 \\ 0 & 1 & 0 & 0 & OH & 0 \\ 0 & 1 & 0 & 0 & 0 & CH3 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \\ 6 \\ 5 \\ 1 \\ 1 \end{pmatrix}$$

$$(1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)$$

fragmentGraph

setGroupReference

connection

The final fragment graph is a matrix with “k” lines and “k” columns.

10.4.4 Dead End case

If no path leads to a fragment, then the group vector is reconsidered. If no group vector leads to a solution, then the parameter m and then the parameter k are reconsidered. All the possibilities are then checked. If in spite of that no possible solution can be found, the search stops and the user is asked to change the parameters of the fragment.

10.5 COMPOSITION HANDLING

10.5.1 Initialization method

The composition of the different elements of the mixture can either be fixed, defined in a range or free. With the different values that are set by the user, the real range of modification of the composition will be calculated. This allows being always sure that at the end the sum of the composition will be equal to 1.

The attribution of the composition is done iteratively. At each iteration, a randomly chosen element gets a composition value. This value is determined by randomly choosing an appropriate integer value "i". The composition value will then be equal to $minValue + i * variation\ step$. Then the real range is updated and the next iteration is considered until a composition value is defined for each element of the mixture.

Let us consider the following example. The user defines a mixture with 4 elements. The first element has a fixed composition of 0.2. The second element has a composition value between 0.2 and 0.9. The two last elements have free compositions: their value is between 0 and 1. The variation step is 0.1.

The constraint information is put into two constraint vectors (Min and Max). As the sum of the compositions must be equal to 1, each constraint is re-evaluated considering:

$$MaxConstraint_i = Min(MaxConstraint_i, 1 - \sum_{j \neq i} MinConstraint_i)$$

$$MinConstraint_i = Max(MinConstraint_i, 1 - \sum_{j \neq i} MaxConstraint_i)$$

User Constraints

TYPE	Min	Max
Fixed	0.2	0.2
Range	0.2	0.9
Free	0	1
Free	0	1

Real Constraints

Min	Max
0.2	0.2
0.2	0.8
0	0.6
0	0.6

Figure 98: Calculation of the real constraints

If we consider the upper constraints of a free composition, even if all the other elements have a composition equal to their minimum value, the composition will never be able to be equal to1 but only to 0.6. Thus 0.6 is the real upper bound of the composition of this element considering the constraints of the other elements of the mixture.

The modification of the constraints is performed in the order of the elements and is repeated until the constraints are stable.

Once the constraints are stable, the algorithm chooses randomly an element to fix its composition and select randomly a value between its upper and lower constraints. This value is then put into the real constraints vectors and the vectors are re-evaluated following the same process as previously.

Choice of an element

Min	Max
0.2	0.2
0.2	0.8
0	0.6
0	0.6

Choice of a value

Min	Max
0.2	0.2
0.2	0.8
0.4	0.4
0	0.6

New real Constraints

Min	Max
0.2	0.2
0.2	0.4
0.4	0.4
0	0.2

Figure 99: Random choice of a composition value

In the example presented on Figure 99, the third element is chosen. Its value must be between 0 and 0.6. As the step of variation is 0.1, the value of the integer “i” is randomly chosen between 0 and 6. Here 4 is chosen. Thus, the value of the composition of the third element is 0.4 (0+4*0.1). Then the real constraints are reevaluated.

This step is repeated until the sum of one of the real constraint is equal to 1. This vector will then be the composition vector of the mixture.

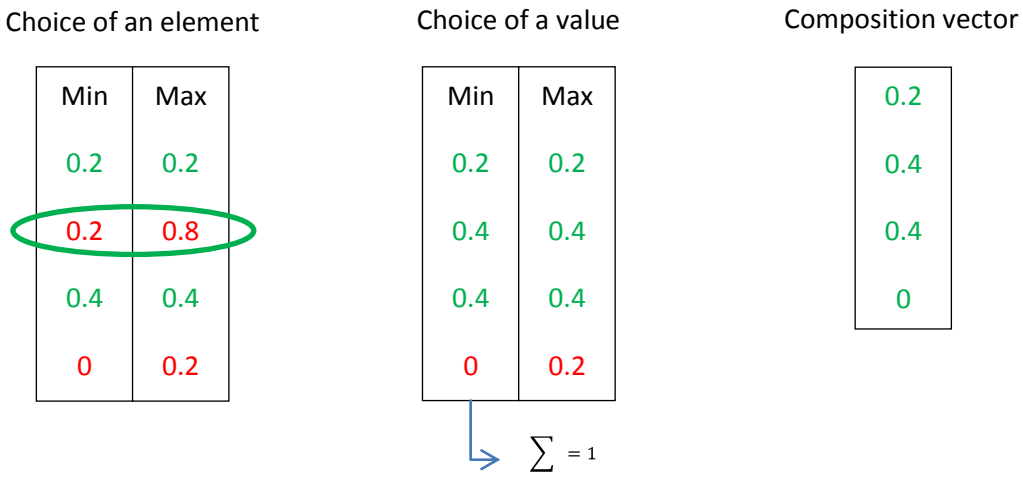


Figure 100: Random choice of a composition value that finalize the determination of the composition vector

In the example presented on Figure 100, the composition of the second element is chosen. Its value must be between 0.2 and 0.8. As the step of variation is 0.1, the value of the integer “i” is randomly chosen between 0 and 4. Here 2 is chosen. Thus, the value of the composition of the second element is 0.4 (0.2+2*0.1). Once this composition value is chosen, it can be observed that the sum of the lower constraints is equal to 1. The composition is necessarily equal to the lower constraints.

10.5.2 Modification method

The modification of the composition starts with the random selection of the element whose composition will be changed. Then, a “direction of modification” (plus or minus) is chosen. Considering the bounds specified by the user and the constraints of the composition values of the other mixtures element, a “quantity of modification”, in terms of number of step variation, is randomly chosen. A second element (or more) is randomly chosen in order to compensate the modification and to keep the sum of the composition equal to 1. Finally, the composition of the elements chosen are modified in accordance with the number of step and with the “direction of modification” chosen

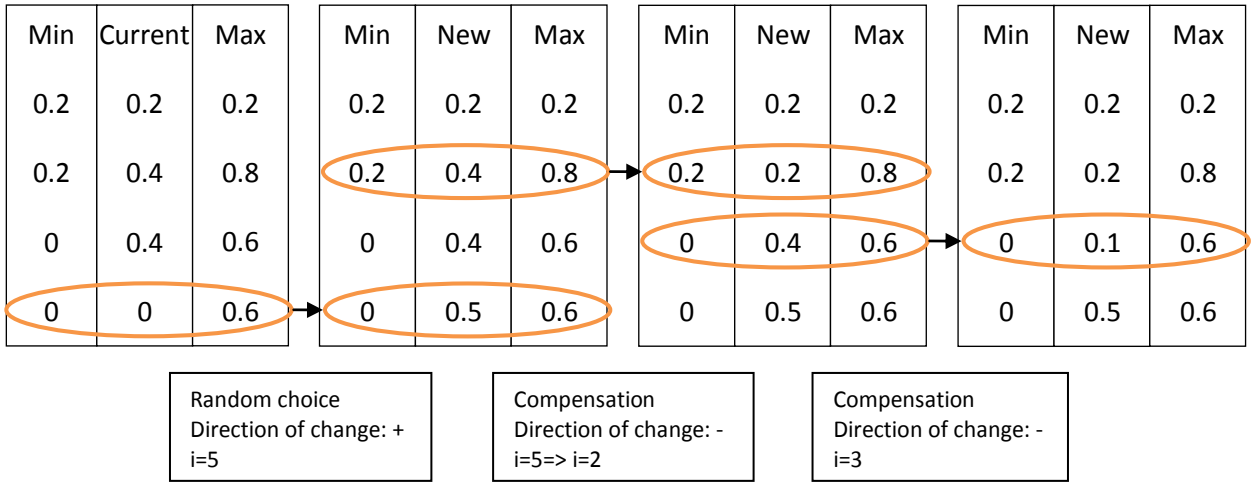


Figure 101: example of a composition modification

In the example presented on the Figure 101, the fourth element is chosen for modification. It is chosen that the composition value of this element will increase of 5 variation step. Thus is new value is 0.5. In order to maintain the sum of the composition equal to 1, another element is selected; its value must decrease of 5 variation steps. The second element is chosen but the lower bound of this element only allows a decrease of 2 variation steps. Another element is selected and its value is decreased of the remaining 3 variation steps.

10.6 IBSS XML INPUT FILE

```
<?xml version="1.0" encoding="utf-8"?>
<PbProblem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <AlgorithmeData>
    <Elitism>30</Elitism>
  </AlgorithmeData>
  <MixtureData>
    <ElementsData>
      <PbElement>
        <Id>1</Id>
        <Molecule xsi:type="PbFreeMolecule">
          <MoleculeSelectionProb>1</MoleculeSelectionProb>
          <Fragments>
            <PbFragment xsi:type="PbListFragment">
              <FragmentSelectionProb>5</FragmentSelectionProb>
              <FragmentGraphList>
                <PbFixedFragment>
                  <StrFragmentGraph>260000</StrFragmentGraph>
                  <StrConnexionVector>a_1</StrConnexionVector>
                </PbFixedFragment>
                <PbFixedFragment>
                  <StrFragmentGraph>260100</StrFragmentGraph>
                  <StrConnexionVector>a_1</StrConnexionVector>
                </PbFixedFragment>
                <PbFixedFragment>
```

```
        <StrFragmentGraph>260300</StrFragmentGraph>
        <StrConnexionVector>a_1</StrConnexionVector>
      </PbFixedFragment>
      <PbFixedFragment>
        <StrFragmentGraph>260500</StrFragmentGraph>
        <StrConnexionVector>a_1</StrConnexionVector>
      </PbFixedFragment>
      <PbFixedFragment>
        <StrFragmentGraph>260600</StrFragmentGraph>
        <StrConnexionVector>a_1</StrConnexionVector>
      </PbFixedFragment>
      <PbFixedFragment>
        <StrFragmentGraph>260700</StrFragmentGraph>
        <StrConnexionVector>a_1</StrConnexionVector>
      </PbFixedFragment>
      <PbFixedFragment>
        <StrFragmentGraph>260800</StrFragmentGraph>
        <StrConnexionVector>a_1</StrConnexionVector>
      </PbFixedFragment>
      <PbFixedFragment>
        <StrFragmentGraph>261100</StrFragmentGraph>
        <StrConnexionVector>a_1</StrConnexionVector>
      </PbFixedFragment>
    </FragmentGraphList>
  </PbFragment>
  <PbFragment xsi:type="PbFreeFragment">
    <FragmentSelectionProb>5</FragmentSelectionProb>
    <Kmax>10</Kmax>
    <ListM>
      <int>1</int>
    </ListM>
    <FunctionalGroupsSet>
      <int>106103</int>
      <int>106102</int>
      <int>106101</int>
      <int>106100</int>
      <int>106202</int>
      <int>106201</int>
      <int>106200</int>
      <int>106301</int>
      <int>106300</int>
      <int>106400</int>
      <int>107102</int>
      <int>107101</int>
      <int>107100</int>
      <int>107201</int>
      <int>107200</int>
      <int>107300</int>
      <int>108101</int>
      <int>108100</int>
      <int>108200</int>
      <int>200000</int>
      <int>200100</int>
      <int>200200</int>
      <int>200300</int>
      <int>200400</int>
      <int>200500</int>
      <int>200600</int>
      <int>200700</int>
      <int>200800</int>
      <int>200900</int>
      <int>201000</int>
      <int>201100</int>
      <int>201200</int>
```

```
            <int>201300</int>
          </FunctionalGroupsSet>
          <CrossOverSelectionProb>65</CrossOverSelectionProb>
          <MutationSelectionProb>15</MutationSelectionProb>
          <InsertionSelectionProb>10</InsertionSelectionProb>
          <DeletionSelectionProb>10</DeletionSelectionProb>
          <ListMInsertion>
            <int>1</int>
          </ListMInsertion>
        </PbFragment>
      </Fragments>
      <StStructure>1 a_1 / a_1 2</StStructure>
    </Molecule>
    <Composition xsi:type="PbFreeComposition">
      <CompElement>1E-10</CompElement>
      <CompSelectionProb>5</CompSelectionProb>
    </Composition>
  </PbElement>
  <PbElement>
    <Id>2</Id>
    <Molecule xsi:type="PbFixedMolecule">
      <StrMolecularGraph>108002</StrMolecularGraph>
    </Molecule>
    <Composition xsi:type="PbFramedComposition">
      <CompElement>1E-10</CompElement>
      <CompSelectionProb>5</CompSelectionProb>
      <MinValue>0.3</MinValue>
      <MaxValue>1</MaxValue>
    </Composition>
  </PbElement>
</ElementsData>
<OpCondtionsData />
<MoleculeModificationProb>9</MoleculeModificationProb>
<CompostionModificationProb>1</CompostionModificationProb>
<OpConditionModificationProb>0</OpConditionModificationProb>
<Step_StaticComposition>0.01</Step_StaticComposition>
<SumEpsilon_StaticComposition>1E-08</SumEpsilon_StaticComposition>
</MixtureData>
<LevelData>
  <PbLevel>
    <IdLevel>1</IdLevel>
    <PopulationSize>100</PopulationSize>
    <StopCriterion>it_300</StopCriterion>
    <ObjectiveFunctionData>
      <Properties>
        <PbProperty xsi:type="PbCalculableProperty">
          <Name>Molecular Weight</Name>
          <Weighting>1</Weighting>
          <Target xsi:type="PbConstraintTarget">
            <Unit>g/mol</Unit>
            <TargetValue>200</TargetValue>
            <PerformanceFunction xsi:type="PbGaussianFunction">
              <Tolerance>20</Tolerance>
              <CorrespondingPerf>0.8</CorrespondingPerf>
            </PerformanceFunction>
            <IsMaximisation>false</IsMaximisation>
          </Target>
          <Model>
            <Identifier>101_MT2010</Identifier>
            <DefaultValueAuthorization>true</DefaultValueAuthorization>
            <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
            <SpecificOpConditionsData />
```

```
        </Model>
        <ConcernedMolecule>
          <int>1</int>
          <int>0</int>
        </ConcernedMolecule>
      </PbProperty>
      <PbProperty xsi:type="PbCalculableProperty">
        <Name>Flash Point</Name>
        <Weighting>1</Weighting>
        <Target xsi:type="PbConstraintTarget">
          <Unit>K</Unit>
          <TargetValue>323.15</TargetValue>
          <PerformanceFunction xsi:type="PbGaussianFunction">
            <Tolerance>5</Tolerance>
            <CorrespondingPerf>0.8</CorrespondingPerf>
          </PerformanceFunction>
          <IsMaximisation>true</IsMaximisation>
        </Target>
        <Model>
          <Identifier>112_CPN2006</Identifier>
          <DefaultValueAuthorization>true</DefaultValueAuthorization>
          <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
          <SpecificOpConditionsData />
        </Model>
        <ConcernedMolecule>
          <int>1</int>
          <int>0</int>
        </ConcernedMolecule>
      </PbProperty>
      <PbProperty xsi:type="PbCalculableProperty">
        <Name>Vapor Pressure</Name>
        <Weighting>1</Weighting>
        <Target xsi:type="PbConstraintTarget">
          <Unit>bar</Unit>
          <TargetValue>0.00267</TargetValue>
          <PerformanceFunction xsi:type="PbGaussianFunction">
            <Tolerance>0.0001</Tolerance>
            <CorrespondingPerf>0.8</CorrespondingPerf>
          </PerformanceFunction>
          <IsMaximisation>false</IsMaximisation>
        </Target>
        <Model>
          <Identifier>111_Riedel1954</Identifier>
          <DefaultValueAuthorization>true</DefaultValueAuthorization>
          <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
          <SpecificOpConditionsData>
            <PbFixedOpCondition>
              <Name>Temp_op</Name>
              <Unit>K</Unit>
              <OpCondvalue>298.15</OpCondvalue>
            </PbFixedOpCondition>
          </SpecificOpConditionsData>
        </Model>
        <ConcernedMolecule>
          <int>1</int>
          <int>0</int>
        </ConcernedMolecule>
      </PbProperty>
    <PbProperty xsi:type="PbCalculableProperty">
        <Name>RED</Name>
        <Weighting>4</Weighting>
    <Target xsi:type="PbConstraintTarget">
          <Unit />
          <TargetValue>0</TargetValue>
```

```
      <PerformanceFunction xsi:type="PbGaussianFunction">
        <Tolerance>1</Tolerance>
        <CorrespondingPerf>0.9</CorrespondingPerf>
      </PerformanceFunction>
      <IsMaximisation>false</IsMaximisation>
    </Target>
    <Model>
      <Identifier>132_MB2010</Identifier>
      <DefaultValueAuthorization>true</DefaultValueAuthorization>
      <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
      <SpecificOpConditionsData>
        <PbFixedOpCondition>
          <Name>HSPd_target</Name>
          <Unit />
          <OpCondvalue>19.7</OpCondvalue>
        </PbFixedOpCondition>
        <PbFixedOpCondition>
          <Name>HSPp_target</Name>
          <Unit />
          <OpCondvalue>11.6</OpCondvalue>
        </PbFixedOpCondition>
        <PbFixedOpCondition>
          <Name>HSPh_target</Name>
          <Unit />
          <OpCondvalue>14.6</OpCondvalue>
        </PbFixedOpCondition>
        <PbFixedOpCondition>
          <Name>HSPradius_target</Name>
          <Unit />
          <OpCondvalue>12.7</OpCondvalue>
        </PbFixedOpCondition>
      </SpecificOpConditionsData>
    </Model>
    <ConcernedMolecule>
      <int>1</int>
      <int>1</int>
    </ConcernedMolecule>
  </PbProperty>
  <PbProperty xsi:type="PbCalculableProperty">
    <Name>Env. Waste</Name>
    <Weighting>0.2</Weighting>
    <Target xsi:type="PbConstraintTarget">
      <Unit />
      <TargetValue>8</TargetValue>
      <PerformanceFunction xsi:type="PbGaussianFunction">
        <Tolerance>1</Tolerance>
        <CorrespondingPerf>0.8</CorrespondingPerf>
      </PerformanceFunction>
      <IsMaximisation>true</IsMaximisation>
    </Target>
    <Model>
      <Identifier>124_Weis2009</Identifier>
      <DefaultValueAuthorization>true</DefaultValueAuthorization>
      <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
      <SpecificOpConditionsData />
    </Model>
    <ConcernedMolecule>
      <int>1</int>
      <int>0</int>
    </ConcernedMolecule>
  </PbProperty>
  <PbProperty xsi:type="PbCalculableProperty">
```

```
      <Name>Env. Impact</Name>
      <Weighting>0.2</Weighting>
      <Target xsi:type="PbConstraintTarget">
        <Unit />
        <TargetValue>8</TargetValue>
        <PerformanceFunction xsi:type="PbGaussianFunction">
          <Tolerance>1</Tolerance>
          <CorrespondingPerf>0.8</CorrespondingPerf>
        </PerformanceFunction>
        <IsMaximisation>true</IsMaximisation>
      </Target>
      <Model>
        <Identifier>125_Weis2009</Identifier>
        <DefaultValueAuthorization>true</DefaultValueAuthorization>
        <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
        <SpecificOpConditionsData />
      </Model>
      <ConcernedMolecule>
        <int>1</int>
        <int>0</int>
      </ConcernedMolecule>
    </PbProperty>
    <PbProperty xsi:type="PbCalculableProperty">
      <Name>Health</Name>
      <Weighting>0.2</Weighting>
      <Target xsi:type="PbConstraintTarget">
        <Unit />
        <TargetValue>8</TargetValue>
        <PerformanceFunction xsi:type="PbGaussianFunction">
          <Tolerance>1</Tolerance>
          <CorrespondingPerf>0.8</CorrespondingPerf>
        </PerformanceFunction>
        <IsMaximisation>true</IsMaximisation>
      </Target>
      <Model>
        <Identifier>126_Weis2009</Identifier>
        <DefaultValueAuthorization>true</DefaultValueAuthorization>
        <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
        <SpecificOpConditionsData />
      </Model>
      <ConcernedMolecule>
        <int>1</int>
        <int>0</int>
      </ConcernedMolecule>
    </PbProperty>
    <PbProperty xsi:type="PbCalculableProperty">
      <Name>Safety</Name>
      <Weighting>0.2</Weighting>
      <Target xsi:type="PbConstraintTarget">
        <Unit />
        <TargetValue>8</TargetValue>
        <PerformanceFunction xsi:type="PbGaussianFunction">
          <Tolerance>1</Tolerance>
          <CorrespondingPerf>0.8</CorrespondingPerf>
        </PerformanceFunction>
        <IsMaximisation>true</IsMaximisation>
      </Target>
      <Model>
        <Identifier>127_Weis2009</Identifier>
        <DefaultValueAuthorization>true</DefaultValueAuthorization>
        <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
        <SpecificOpConditionsData />
      </Model>
      <ConcernedMolecule>
```

```
      <int>1</int>
      <int>0</int>
    </ConcernedMolecule>
  </PbProperty>
  <PbProperty xsi:type="PbCalculableProperty">
    <Name>LCA</Name>
    <Weighting>0.2</Weighting>
    <Target xsi:type="PbConstraintTarget">
      <Unit />
      <TargetValue>8</TargetValue>
      <PerformanceFunction xsi:type="PbGaussianFunction">
        <Tolerance>1</Tolerance>
        <CorrespondingPerf>0.8</CorrespondingPerf>
      </PerformanceFunction>
      <IsMaximisation>true</IsMaximisation>
    </Target>
    <Model>
      <Identifier>128_Weis2009</Identifier>
      <DefaultValueAuthorization>true</DefaultValueAuthorization>
      <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
      <SpecificOpConditionsData />
    </Model>
    <ConcernedMolecule>
      <int>1</int>
      <int>0</int>
    </ConcernedMolecule>
  </PbProperty>
  <PbProperty xsi:type="PbCalculableProperty">
    <Name>Viscosity (300K)</Name>
    <Weighting>1</Weighting>
    <Target xsi:type="PbFrameTarget">
      <Unit>cp</Unit>
      <TargetValue>0.8</TargetValue>
      <PerformanceFunction xsi:type="PbGaussianFunction">
        <Tolerance>0.1</Tolerance>
        <CorrespondingPerf>0.8</CorrespondingPerf>
      </PerformanceFunction>
      <SecondValue>1.4</SecondValue>
      <SecondPerformanceFunction xsi:type="PbGaussianFunction">
        <Tolerance>0.1</Tolerance>
        <CorrespondingPerf>0.8</CorrespondingPerf>
      </SecondPerformanceFunction>
    </Target>
    <Model>
      <Identifier>121_CMMG2008</Identifier>
      <DefaultValueAuthorization>true</DefaultValueAuthorization>
      <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
      <SpecificOpConditionsData>
        <PbFixedOpCondition>
        <Name>Temp_op</Name>
        <Unit>K</Unit>
        <OpCondvalue>298.15</OpCondvalue>
        </PbFixedOpCondition>
      </SpecificOpConditionsData>
    </Model>
    <ConcernedMolecule>
      <int>1</int>
      <int>1</int>
    </ConcernedMolecule>
  </PbProperty>
  <PbProperty xsi:type="PbCalculableProperty">
    <Name>Superfical Tension (298K)</Name>
```

```
      <Weighting>1</Weighting>
      <Target xsi:type="PbFrameTarget">
        <Unit>dyn/cm2</Unit>
        <TargetValue>30</TargetValue>
        <PerformanceFunction xsi:type="PbGaussianFunction">
          <Tolerance>5</Tolerance>
          <CorrespondingPerf>0.8</CorrespondingPerf>
        </PerformanceFunction>
        <SecondValue>45</SecondValue>
        <SecondPerformanceFunction xsi:type="PbGaussianFunction">
          <Tolerance>5</Tolerance>
          <CorrespondingPerf>0.8</CorrespondingPerf>
        </SecondPerformanceFunction>
      </Target>
      <Model>
        <Identifier>122_CMMG2008</Identifier>
        <DefaultValueAuthorization>true</DefaultValueAuthorization>
        <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
        <SpecificOpConditionsData />
      </Model>
      <ConcernedMolecule>
        <int>1</int>
        <int>1</int>
      </ConcernedMolecule>
    </PbProperty>
    <PbProperty xsi:type="PbCalculableProperty">
      <Name>Log(Ws)</Name>
      <Weighting>4</Weighting>
      <Target xsi:type="PbConstraintTarget">
        <Unit>mg/L</Unit>
        <TargetValue>4</TargetValue>
        <PerformanceFunction xsi:type="PbGaussianFunction">
          <Tolerance>0.5</Tolerance>
          <CorrespondingPerf>0.8</CorrespondingPerf>
        </PerformanceFunction>
        <IsMaximisation>true</IsMaximisation>
      </Target>
      <Model>
        <Identifier>131_MG2002</Identifier>
        <DefaultValueAuthorization>true</DefaultValueAuthorization>
        <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
        <SpecificOpConditionsData />
      </Model>
      <ConcernedMolecule>
        <int>1</int>
        <int>0</int>
      </ConcernedMolecule>
    </PbProperty>
    <PbProperty xsi:type="PbCalculableProperty">
      <Name>Density</Name>
      <Weighting>1</Weighting>
      <Target xsi:type="PbFrameTarget">
        <Unit>noUnit</Unit>
        <TargetValue>0.9</TargetValue>
        <PerformanceFunction xsi:type="PbGaussianFunction">
          <Tolerance>0.05</Tolerance>
          <CorrespondingPerf>0.8</CorrespondingPerf>
        </PerformanceFunction>
        <SecondValue>1.1</SecondValue>
        <SecondPerformanceFunction xsi:type="PbGaussianFunction">
          <Tolerance>0.05</Tolerance>
          <CorrespondingPerf>0.8</CorrespondingPerf>
        </SecondPerformanceFunction>
      </Target>
```

```
      <Model>
        <Identifier>119_MB2010</Identifier>
        <DefaultValueAuthorization>true</DefaultValueAuthorization>
        <ModelSubstitutionAuthorization>true</ModelSubstitutionAuthorization>
        <SpecificOpConditionsData />
      </Model>
      <ConcernedMolecule>
        <int>1</int>
        <int>1</int>
      </ConcernedMolecule>
    </PbProperty>
      </Properties>
      <Penalization>0.3</Penalization>
      <DefaultOpConditions>
        <PbFixedOpCondition>
          <Name>Temp_op</Name>
          <Unit>K</Unit>
          <OpCondvalue>298.15</OpCondvalue>
        </PbFixedOpCondition>
        <PbFixedOpCondition>
          <Name>Pressure_op</Name>
          <Unit>bar</Unit>
          <OpCondvalue>101325</OpCondvalue>
        </PbFixedOpCondition>
        <PbFixedOpCondition>
          <Name>HeatQ_op</Name>
          <Unit>kJ</Unit>
          <OpCondvalue>0</OpCondvalue>
        </PbFixedOpCondition>
        <PbFixedOpCondition>
          <Name>Vap_ratio</Name>
          <Unit />
          <OpCondvalue>0</OpCondvalue>
        </PbFixedOpCondition>
        <PbFixedOpCondition>
          <Name>LLsplit_ratio</Name>
          <Unit />
          <OpCondvalue>0</OpCondvalue>
        </PbFixedOpCondition>
      </DefaultOpConditions>
    </ObjectiveFunctionData>
    <IterationWherePopSaved>
      <int>10</int>
      <int>20</int>
      <int>30</int>
      <int>40</int>
      <int>50</int>
      <int>60</int>
      <int>70</int>
      <int>80</int>
      <int>90</int>
      <int>100</int>
      <int>150</int>
      <int>200</int>
      <int>250</int>
    </IterationWherePopSaved>
  </PbLevel>
 </LevelData>
</PbProblem>
```

10.7 IBSS OUTPUT FILE EXTRACT

N°1
performance= 0.959492806122624
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.529271892056647
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.40672219236265cp
Superfical Tension (298K): 44.6387213513382dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03390074787295noUnit
Molecule N°1
Composition = 0.31
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.69
water
N°2
performance= 0.959483072090784
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.531870156002891
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.4010758764631cp
Superfical Tension (298K): 44.63872038434dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.0344315558975noUnit
Molecule N°1
Composition = 0.32
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.68
water
N°3
performance= 0.959404814271987
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.52737726843303
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391

Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.41188490739863cp
Superfical Tension (298K): 44.6387223828031dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03334469931804noUnit
Molecule N°1
Composition = 0.3
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.7
water
N°4
performance= 0.959391899563443
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.535063368739207
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.39498925407358cp
Superfical Tension (298K): 44.6387194759475dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03493880385406noUnit
Molecule N°1
Composition = 0.33
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.67
water
N°5
performance= 0.959283824620189
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.538756016762459
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.38850235514826cp
Superfical Tension (298K): 44.6387186209899dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03542402626812noUnit
Molecule N°1
Composition = 0.34
BMS3 1
1 alcenyle 1 0 0

1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.66
water
N°6
performance= 0.959240699292394
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.526309332517178
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.41651719134954cp
Superfical Tension (298K): 44.6387234854033dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.0327615660623noUnit
Molecule N°1
Composition = 0.29
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.71
water
N°7
performance= 0.959162757780026
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.54286454992848
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.38165219546845cp
Superfical Tension (298K): 44.638717814887dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03588862723369noUnit
Molecule N°1
Composition = 0.35
BMS3 1
CH3- 0 1 0
1 0 alcenyle 1 0
1 1 -CH-- 1
0 0 1 CH3-
Molecule N°2
Composition = 0.65
water
N°8
performance= 0.959030607534105
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar

RED: 0.547316176275308
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.37447300725794cp
Superfical Tension (298K): 44.6387170535675dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03633389398296noUnit
Molecule N°1
Composition = 0.36
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.64
water

N°9
performance= 0.959021877385793
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.526206571300025
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.42056838392516cp
Superfical Tension (298K): 44.6387246667605dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03214931979431noUnit
Molecule N°1
Composition = 0.28
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.72
water

N°10
performance= 0.958889040116762
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.552047700997809
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.36699645164382cp
Superfical Tension (298K): 44.6387163334004dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03676100879641noUnit

Molecule N°1
Composition = 0.37
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.63
water
N°11
performance= 0.958768789122598
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.527223836342879
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.42398367610473cp
Superfical Tension (298K): 44.6387259356256dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03150572449014noUnit
Molecule N°1
Composition = 0.27
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.73
water
N°12
performance= 0.958739511820086
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.557004433353165
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.35925181435753cp
Superfical Tension (298K): 44.6387156511367dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03717105948536noUnit
Molecule N°1
Composition = 0.38
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.62
water
N°13

performance= 0.958583296646805
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.562139176028956
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.35126618598463cp
Superfical Tension (298K): 44.6387150038609dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03756504864285noUnit
Molecule N°1
Composition = 0.39
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.61
water

N°14
performance= 0.958500036118482
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.529533044728017
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.42670376538051cp
Superfical Tension (298K): 44.6387273020956dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03082830912579noUnit
Molecule N°1
Composition = 0.26
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.74
water

N°15
performance= 0.958421510029431
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.56741130416805
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732

Viscosity (300K): 1.34306462798516cp
Superfical Tension (298K): 44.6387143889489dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.0379439018282noUnit
Molecule N°1
Composition = 0.4
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.6
water
N°16
performance= 0.958255129221279
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.572785935761503
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.33467032562012cp
Superfical Tension (298K): 44.6387138040326dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03830847482615noUnit
Molecule N°1
Composition = 0.41
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.59
water
N°17
performance= 0.958231403597014
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.533323601602437
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.42866448329453cp
Superfical Tension (298K): 44.6387287778831dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03011433597186noUnit
Molecule N°1
Composition = 0.25
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-

Molecule N°2
Composition = 0.75
water
N°18
performance= 0.958085010864596
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.578233191261477
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.32610472883676cp
Superfical Tension (298K): 44.6387132469694dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03865956010033noUnit
Molecule N°1
Composition = 0.42
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.58
water
N°19
performance= 0.95798028623576
Molecular Weight: 192.214g/mol
Flash Point: 368.032916758153K
Vapor Pressure: 2.84280042515238E-05bar
RED: 0.538206451107371
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.40182936142585cp
Superfical Tension (298K): 46.4938048361451dyn/cm2
Log(Ws): 4.53128999999999mg/L
Density: 1.04102467114549noUnit
Molecule N°1
Composition = 0.32
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 vinyle
Molecule N°2
Composition = 0.68
water
N°20
performance= 0.957974706502363
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.538802516439778
Env. Waste: 8.1283

Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.42979639376516cp
Superfical Tension (298K): 44.6387303766526dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.0293607636127noUnit
Molecule N°1
Composition = 0.24
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.76
water

N°21
performance= 0.957954843006126
Molecular Weight: 192.214g/mol
Flash Point: 368.032916758153K
Vapor Pressure: 2.84280042515238E-05bar
RED: 0.539242599935837
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.39593436543464cp
Superfical Tension (298K): 46.4938046206306dyn/cm2
Log(Ws): 4.53128999999999mg/L
Density: 1.04156094801898noUnit
Molecule N°1
Composition = 0.33
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 vinyle
Molecule N°2
Composition = 0.67
water

N°22
performance= 0.957918807934406
Molecular Weight: 192.214g/mol
Flash Point: 368.032916758153K
Vapor Pressure: 2.84280042515238E-05bar
RED: 0.537769316424353
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.40730211122392cp
Superfical Tension (298K): 46.4938050655637dyn/cm2
Log(Ws): 4.53128999999999mg/L
Density: 1.040463043227noUnit
Molecule N°1
Composition = 0.31
BMS3 1

1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 vinyle
Molecule N°2
Composition = 0.69
water
N°23
performance= 0.957911906158439
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.583727537732597
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.31738768208552cp
Superfical Tension (298K): 44.6387127158161dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03899789254339noUnit
Molecule N°1
Composition = 0.43
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.57
water
N°24
performance= 0.957909166424576
Molecular Weight: 192.214g/mol
Flash Point: 368.032916758153K
Vapor Pressure: 2.84280042515238E-05bar
RED: 0.540794827407064
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.38965548821527cp
Superfical Tension (298K): 46.4938044177935dyn/cm2
Log(Ws): 4.53128999999999mg/L
Density: 1.04207355242567noUnit
Molecule N°1
Composition = 0.34
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 vinyle
Molecule N°2
Composition = 0.66
water
N°25
performance= 0.95785028306023
Molecular Weight: 192.214g/mol

Flash Point: 368.032916758153K
Vapor Pressure: 2.84280042515238E-05bar
RED: 0.542789709819585
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.38302813902787cp
Superfical Tension (298K): 46.4938042265471dyn/cm2
Log(Ws): 4.53128999999999mg/L
Density: 1.04256401795143noUnit
Molecule N°1
Composition = 0.35
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 vinyle 0
0 1 0 CH3-
Molecule N°2
Composition = 0.65
water

N°26
performance= 0.957779982125267
Molecular Weight: 192.214g/mol
Flash Point: 368.032916758153K
Vapor Pressure: 2.84280042515238E-05bar
RED: 0.545162431480045
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.37608500795629cp
Superfical Tension (298K): 46.4938040459254dyn/cm2
Log(Ws): 4.53128999999999mg/L
Density: 1.04303374851414noUnit
Molecule N°1
Composition = 0.36
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 vinyle 0
0 1 0 CH3-
Molecule N°2
Composition = 0.64
water

N°27
performance= 0.95776551217358
Molecular Weight: 192.214g/mol
Flash Point: 368.032916758153K
Vapor Pressure: 2.84280042515238E-05bar
RED: 0.538024574708778
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.41231103571427cp
Superfical Tension (298K): 46.4938053102769dyn/cm2
Log(Ws): 4.53128999999999mg/L

Density: 1.03987422314294noUnit
Molecule N°1
Composition = 0.3
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 vinyle
Molecule N°2
Composition = 0.7
water
N°28
performance= 0.957736473980802
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.589247211352263
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.30853754396592cp
Superfical Tension (298K): 44.6387122088061dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03932415461184noUnit
Molecule N°1
Composition = 0.44
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.56
water
N°29
performance= 0.957736370053276
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.546194230397459
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.43002436090125cp
Superfical Tension (298K): 44.6387321144454dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.0285642036369noUnit
Molecule N°1
Composition = 0.23
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.77

water
N°30
performance= 0.957699828823528
Molecular Weight: 192.214g/mol
Flash Point: 368.032916758153K
Vapor Pressure: 2.84280042515238E-05bar
RED: 0.54785592249444
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.36885628241343cp
Superfical Tension (298K): 46.4938038750671dyn/cm2
Log(Ws): 4.53128999999999mg/L
Density: 1.04348403178459noUnit
Molecule N°1
Composition = 0.37
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 vinyle 0
0 1 0 CH3-
Molecule N°2
Composition = 0.63
water
N°31
performance= 0.957611194274631
Molecular Weight: 192.214g/mol
Flash Point: 368.032916758153K
Vapor Pressure: 2.84280042515238E-05bar
RED: 0.550820048168085
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.36136984536565cp
Superfical Tension (298K): 46.4938037132013dyn/cm2
Log(Ws): 4.53128999999999mg/L
Density: 1.04391605097436noUnit
Molecule N°1
Composition = 0.38
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 vinyle 0
0 1 0 CH3-
Molecule N°2
Composition = 0.62
water
N°32
performance= 0.957559292263103
Molecular Weight: 180.203g/mol
Flash Point: 356.454776773145K
Vapor Pressure: 7.47383874841577E-05bar
RED: 0.594773711319794
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675

LCA: 4.3732
Viscosity (300K): 1.29957129752761cp
Superfical Tension (298K): 44.6387117243299dyn/cm2
Log(Ws): 4.48283499999999mg/L
Density: 1.03963898092083noUnit
Molecule N°1
Composition = 0.45
BMS3 1
1 alcenyle 1 0 0
1 -CH-- 1 1
0 1 CH3- 0
0 1 0 CH3-
Molecule N°2
Composition = 0.55
water
N°33
performance= 0.956774825148985
Molecular Weight: 152.149g/mol
Flash Point: 336.552420068399K
Vapor Pressure: 0.00023991926800688bar
RED: 0.450757826162366
Env. Waste: 4.9799
Env. Impact: 6.5002
Health: 7.7391
Safety: 5.9947
LCA: 4.3732
Viscosity (300K): 1.25022630619606cp
Superfical Tension (298K): 44.6518688956472dyn/cm2
Log(Ws): 4.682975mg/L
Density: 1.08175638712681noUnit
Molecule N°1
Composition = 0.42
BMS3 1
1 alcenyle 1
1 CH3-
Molecule N°2
Composition = 0.58
water
N°34
performance= 0.956769128316393
Molecular Weight: 152.149g/mol
Flash Point: 336.552420068399K
Vapor Pressure: 0.00023991926800688bar
RED: 0.450987549321481
Env. Waste: 4.9799
Env. Impact: 6.5002
Health: 7.7391
Safety: 5.9947
LCA: 4.3732
Viscosity (300K): 1.23807982611209cp
Superfical Tension (298K): 44.6518581140448dyn/cm2
Log(Ws): 4.682975mg/L
Density: 1.08243650612058noUnit
Molecule N°1
Composition = 0.43
BMS3 1
1 alcenyle 1
1 CH3-
Molecule N°2
Composition = 0.57

water
N°35
performance= 0.956768967583158
Molecular Weight: 152.149g/mol
Flash Point: 336.552420068399K
Vapor Pressure: 0.00023991926800688bar
RED: 0.450994029218998
Env. Waste: 4.9799
Env. Impact: 6.5002
Health: 7.7391
Safety: 5.9947
LCA: 4.3732
Viscosity (300K): 1.26223655347711cp
Superfical Tension (298K): 44.6518802031725dyn/cm2
Log(Ws): 4.682975mg/L
Density: 1.0810521795783noUnit
Molecule N°1
Composition = 0.41
BMS3 1
1 alcenyle 1
1 CH3-
Molecule N°2
Composition = 0.59
water
N°36
performance= 0.956753023788663
Molecular Weight: 152.149g/mol
Flash Point: 336.552420068399K
Vapor Pressure: 0.00023991926800688bar
RED: 0.45163635485741
Env. Waste: 4.9799
Env. Impact: 6.5002
Health: 7.7391
Safety: 5.9947
LCA: 4.3732
Viscosity (300K): 1.22582029718413cp
Superfical Tension (298K): 44.6518478225075dyn/cm2
Log(Ws): 4.682975mg/L
Density: 1.08309375175749noUnit
Molecule N°1
Composition = 0.44
BMS3 1
1 alcenyle 1
1 CH3-
Molecule N°2
Composition = 0.56
water
N°37
performance= 0.956750284283747
Molecular Weight: 152.149g/mol
Flash Point: 336.552420068399K
Vapor Pressure: 0.00023991926800688bar
RED: 0.45174663294802
Env. Waste: 4.9799
Env. Impact: 6.5002
Health: 7.7391
Safety: 5.9947
LCA: 4.3732
Viscosity (300K): 1.27408548309006cp
Superfical Tension (298K): 44.6518920760643dyn/cm2
Log(Ws): 4.682975mg/L

Density: 1.08032258064516noUnit
Molecule N°1
Composition = 0.4
BMS3 1
1 alcenyle 1
1 CH3-
Molecule N°2
Composition = 0.6
water
N°38
performance= 0.95666863970857
Molecular Weight: 152.149g/mol
Flash Point: 336.552420068399K
Vapor Pressure: 0.00023991926800688bar
RED: 0.455021475122067
Env. Waste: 4.9799
Env. Impact: 6.5002
Health: 7.7391
Safety: 5.9947
LCA: 4.3732
Viscosity (300K): 1.29718870073892cp
Superfical Tension (298K): 44.6519176964805dyn/cm2
Log(Ws): 4.682975mg/L
Density: 1.07878150997686noUnit
Molecule N°1
Composition = 0.38
BMS3 1
1 alcenyle 1
1 CH3-
Molecule N°2
Composition = 0.62
water
N°39
performance= 0.955228923062431
Molecular Weight: 164.16g/mol
Flash Point: 351.82549758119K
Vapor Pressure: 8.89329711083451E-05bar
RED: 0.475493665341761
Env. Waste: 4.9799
Env. Impact: 6.5002
Health: 7.7391
Safety: 5.9947
LCA: 4.3732
Viscosity (300K): 1.36083730834699cp
Superfical Tension (298K): 46.2453953890617dyn/cm2
Log(Ws): 4.42421999999999mg/L
Density: 1.07275286480758noUnit
Molecule N°1
Composition = 0.32
BMS3 1
1 alcenyle 1
1 vinyle
Molecule N°2
Composition = 0.68
water
N°40
performance= 0.952245821485876
Molecular Weight: 154.165g/mol
Flash Point: 327.435009084978K
Vapor Pressure: 0.00063303482355071 6bar

RED: 0.531756854589555
Env. Waste: 5.5339
Env. Impact: 6.043
Health: 7.7391
Safety: 5.6375
LCA: 4.3732
Viscosity (300K): 1.4053775820826cp
Superfical Tension (298K): 43.8986083621172dyn/cm2
Log(Ws): 4.94766499999999mg/L
Density: 1.04595456886954noUnit
Molecule N°1
Composition = 0.33
BMS3 1
CH3- 1 0
1 1 -CH-- 1
0 1 CH3-
Molecule N°2
Composition = 0.67
water
N°41
performance= 0.952190609915709
Molecular Weight: 204.225g/mol
Flash Point: 378.481556169258K
Vapor Pressure: 1.1310805527493E-05bar
RED: 0.556069949439751
Env. Waste: 8.1283
Env. Impact: 5.4842
Health: 7.7391
Safety: 5.5675
LCA: 4.3732
Viscosity (300K): 1.39431994586883cp
Superfical Tension (298K): 48.2268010629777dyn/cm2
Log(Ws): 4.53953500000001mg/L
Density: 1.04310083882718noUnit
Molecule N°1
Composition = 0.34
BMS3 1
vinyle 0 1 0
1 0 alcenyle 1 0
1 1 -CH-- 1
0 0 1 vinyle
Molecule N°2
Composition = 0.66
water
N°42
performance= 0.951879771678181
Molecular Weight: 152.149g/mol
Flash Point: 336.552420068399K
Vapor Pressure: 0.00023991926800688bar
RED: 0.610095745921569
Env. Waste: 4.9799
Env. Impact: 6.5002
Health: 7.7391
Safety: 5.9947
LCA: 4.3732
Viscosity (300K): 1.41391888757598cp
Superfical Tension (298K): 44.652251871521dyn/cm2
Log(Ws): 4.682975mg/L
Density: 1.06211571900539noUnit
Molecule N°1
Composition = 0.23

BMS3 1
1 alcenyle 1
1 CH3-
Molecule N°2
Composition = 0.77
water
N°43
performance= 0.951703666800949
Molecular Weight: 166.176g/mol
Flash Point: 340.87208252206K
Vapor Pressure: 0.00021353976749257bar
RED: 0.540253168745294
Env. Waste: 5.5339
Env. Impact: 6.043
Health: 7.7391
Safety: 5.6375
LCA: 4.3732
Viscosity (300K): 1.37946200218475cp
Superfical Tension (298K): 45.7533831289004dyn/cm2
Log(Ws): 4.99611999999999mg/L
Density: 1.0543318163012noUnit
Molecule N°1
Composition = 0.36
BMS3 1
1 -CH-- 1 1
1 vinyle 0
1 0 CH3-
Molecule N°2
Composition = 0.64
water
N°44
performance= 0.951116883176762
Molecular Weight: 166.176g/mol
Flash Point: 340.87208252206K
Vapor Pressure: 0.00021353976749257bar
RED: 0.547744977288275
Env. Waste: 5.5339
Env. Impact: 6.043
Health: 7.7391
Safety: 5.6375
LCA: 4.3732
Viscosity (300K): 1.41568070586769cp
Superfical Tension (298K): 45.7534040786425dyn/cm2
Log(Ws): 4.99611999999999mg/L
Density: 1.05093038161917noUnit
Molecule N°1
Composition = 0.31
BMS3 1
1 -CH-- 1 1
1 CH3- 0
1 0 vinyle
Molecule N°2
Composition = 0.69
water
N°45
performance= 0.941504350493172
Molecular Weight: 138.122g/mol
Flash Point: 317.84396037589K
Vapor Pressure: 0.000888212832108586bar
RED: 0.604022676358798

Env. Waste: 4.4259
Env. Impact: 6.8558
Health: 7.7391
Safety: 6.6391
LCA: 4.3732
Viscosity (300K): 1.41138785787372cp
Superfical Tension (298K): 45.5076007615545dyn/cm2
Log(Ws): 5.38977000000001mg/L
Density: 1.08080427446569noUnit
Molecule N°1
Composition = 0.25
BMS3 1
1 vinyle
Molecule N°2
Composition = 0.75
water

10.8 DELPHI QUESTIONNAIRE

DELPHI selection of a CAMD generated product for laboratory consolidation

Round 1

Foreword:

The questionnaire concerns the subjects presented in the Requirements Reference Document (ref. XX) and in the Set of Alternative Solutions File (ref. YY). It is organised in four distinct groups of questions:

I. Analysis of the theoretical performance of the alternatives
II. Analysis of the structure of the product
III. Analysis of the parameters of the search algorithm
IV. General analysis

The first three groups contain detailed technical questions addressed principally to the experts of the technical fields discussed. However, the opinion of everyone is welcome. The last group of questions is more general and addressed to all the experts of the panel.

I. Analysis of the theoretical performance of the alternatives

1. Do you consider that the theoretical performance of the alternatives has been correctly calculated?

□ Yes □ No □ No opinion

If not:

2. Are you unsatisfied with the properties evaluated?
   a) Do you see any missing property? Which one(s)?
   Note: If yes, please justify, and please also fill the "Property to be added" fields in next sections.

Free answer:

   b) Do you see any unnecessary property? Which one(s)?
   Note: If yes, please justify.

Free answer:

3. Are you unsatisfied with the property targets which have been set?

Property	Former target	New target	Justification (mandatory when target is changed)
Molecular weight	<200g/mol		
Flash point	>323.15K		
Vapor pressure	<0.00267bar		
RED	<0		
Env.Waste	>8		
Env.Impact	>8		
Health	>8		
Safety	>8		
LCA	>8		
Viscosity (300K)	=0.8 cp		
Superficial Tension (298K)	=30 dyn/cm2		
Log(Ws)	>4 mg/L		
Density	=0.9		
Property to be added (if possible, precise the desired target)	N/A		
	N/A		
	N/A		
	N/A		
	N/A		
	N/A		
	N/A		
	N/A		
	N/A		

4. Are you unsatisfied with the weighting of the properties?

Action						
Increase weight considerably	Increase weight moderately	Keep weight identical	Decrease weight moderately	Decrease weight considerably	Suppress property	No opinion
>>	>	=	<	<<	X	/

Property	Weight	Action (>>, >, =, <, <<, X or /)	Justification (mandatory except when action is "=", "X" or "/")
Molecular weight	1		
Flash point	1		
Vapor pressure	1		
RED	4		
Env.Waste	0.2		
Env.Impact	0.2		
Health	0.2		
Safety	0.2		
LCA	0.2		
Viscosity (300K)	1		
Superficial Tension (298K)	1		
Log(Ws)	4		
Density	1		
Property to be added (if possible, precise the desired weight)		N/A	
		N/A	
		N/A	
		N/A	
		N/A	
		N/A	
		N/A	
		N/A	
		N/A	

5. Are you unsatisfied with the performance functions which have been chosen for each property?

Property	Former performance function	New performance function	Justification (mandatory when performance function is changed)
Molecular weight	G(20,0.8)		
Flash point	G(5,0.8)		
Vapor pressure	G(0.0001,0.8)		
RED	G(1,0.9)		
Env.Waste	G(1,0.8)		
Env.Impact	G(1,0.8)		
Health	G(1,0.8)		
Safety	G(1,0.8)		
LCA	G(1,0.8)		
Viscosity (300K)	G(0.1,0.8)		
Superficial Tension (298K)	G(5,0.8)		
Log(Ws)	G(0.5,0.8)		
Density	G(0.05,0.8)		
Property to be added (if possible, precise the desired performance function)	N/A		
	N/A		
	N/A		
	N/A		
	N/A		
	N/A		
	N/A		
	N/A		
	N/A		

6. Are you unsatisfied with the models which have been chosen for evaluating each property?

Property	Former		New		Justification (mandatory when model, operating conditions or parameters are changed)
	Model	Operating conditions - Parameters	Model	Operating conditions - Parameters	
Molecular weight					
Flash point	Catoire et al., 2006				
Vapor pressure	Riedel, 1954	T=298.15 K			
RED	HSPiP	HSPd=19.7 HSPp=11.6 HSPh=14.6 HSPradius=12.7			
Env.Waste	Weis, 2009				
Env.Impact	Weis, 2009				
Health	Weis, 2009				
Safety	Weis, 2009				
LCA	Weis, 2009				
Viscosity (300K)	Conte, 2008	T=298.15 K			
Superficial Tension (298K)	Conte, 2008				
Log(Ws)	Marerro and Gani, 2002				
Density	HSPiP				
Property to be added (if possible, precise the desired model)	N/A	N/A			
	N/A	N/A			
	N/A	N/A			
	N/A	N/A			
	N/A	N/A			
	N/A	N/A			
	N/A	N/A			
	N/A	N/A			
	N/A	N/A			

)o you have any other concern(s) with the way the theoretical performance of the lternatives has been calculated? Which one(s)?

'ote: If yes, please justify and propose a way forward.

ıver:

II. Analysis of the structure of the product

1. Do you consider that the structure of the product has been correctly constrained?

□ Yes □ No □ No opinion

If not:

2. Are you unsatisfied with the binary structure of the product?

Note: If yes, please justify and propose a way forward.

Free answer:

3. Are you unsatisfied with the first molecule of the mixture (fixed, H_2O)?

a) Are you unsatisfied with the fact that it is fixed?

Note: If yes, please justify and propose a way forward.

Free answer:

b) Are you unsatisfied with the fact that it is H_2O?

Note: If yes, please justify and propose a way forward.

Free answer:

c) Are you unsatisfied with the fact that the composition of the first molecule in the mixture is at least 30%?

Note: If yes, please justify and propose a way forward.

nswer:

re you unsatisfied with the second molecule of the mixture (free, two fragments)?

a) Are you unsatisfied with the fact that it is free?

Note: If yes, please justify and propose a way forward.

nswer:

b) Are you unsatisfied with the fact that it is composed of 2 fragments?

Note: If yes, please justify and propose a way forward.

nswer:

c) Do you consider that the first fragment should be of another type (free, fixed)?

Note: If yes, please justify and propose a way forward.

Free answer:

d) Do you consider that another list of fragments should be considered for the first fragment?

Note: If yes, please justify and propose a way forward.

Free answer:

e) Do you consider that the second fragment should be of another type (from a list, fixed)?

Note: If yes, please justify and propose a way forward.

Free answer:

f) For the second fragment, are you unsatisfied with the maximum number of groups (equal to 10), with the minimum number of groups (equal to 1), or with the absence of cycles?
Note: If yes, please justify and propose a way forward.

answer:

g) Do you consider that another list of groups should be considered for the first fragment?
Note: If yes, please justify and propose a way forward.

answer:

o you have any other concern(s) with the structure of the product? Which one(s)?
ote: If yes, please justify and propose a way forward.

ver:

III. Analysis of the parameters of the search algorithm

1. Do you consider that the parameters of the search algorithm have been correctly set?

□ Yes □ No □ No opinion

If not:

2. Are you unsatisfied with the elitism (30 individuals) or the population size (100 individuals)?
Note: If yes, please justify and propose a way forward.

Free answer:

3. Are you unsatisfied with the number of iterations (300)?
Note: If yes, please justify and propose a way forward.

Free answer:

4. Are you unsatisfied with the probabilities of selection for modification?
Note: If yes, please justify and propose a way forward.

Free answer:

5. Do you have any other concern(s) with the parameters of the search algorithm? Which one(s)?

Note: If yes, please justify and propose a way forward.

Free answer:

IV. General analysis

1. Do you consider that there is a sufficient number of alternatives with a satisfactory theoretical performance?

Judgement (please check off)				
No satisfactory alternative	Small number of satisfactory alternatives	Medium number of satisfactory alternatives	Correct number of satisfactory alternatives	Large number of satisfactory alternatives

Justification / comments:

2. From your perspective :

- ☐ The set of alternatives proposed is worthy of consideration. A particular alternative from this set shall be selected for laboratory consolidation.
- ☐ The set of alternatives proposed is not worthy of consideration, and a better set can be obtained by taking into account the answers given to this questionnaire.
- ☐ The set of alternatives proposed is not worthy of consideration, and no satisfactory set can be obtained.

' you consider that the set of alternatives proposed is worthy of consideration, please lentify the top 5 alternatives which according to you must be considered in priority r laboratory testing:

ernative #1	
ernative #2	
ernative #3	
ernative #4	
ernative #5	

al comments:

4. If you see alternatives which must be rejected, please list them and provide a justification:

Alternative to be rejected	Justification

10.9 ACRONYMS

BPM: Business Process Modeling

BPMN: Business Process Modeling Notation

BRE: Business Rule Engine

BRM: Business Rule Management

BRMS: Business Rules Management System

CAMD: Computer Aided Molecular Design

CAPD: Computer Aided Product Design

CAPE: Computer Aided Process Engineering

DLL: Dynamic Linked Library

DM: Decision Making

DSML: Domain Specific Modeling Languages

DSS: decision support system

ICT: Information and Communication Technology

IDE: Integrated Development Environment

IS: Information System

KBE: Knowledge Based Engineering

MCDA: Multi Criteria Decision Analysis

MCDM: Multi Criteria Decision Making

MDA: Modeling Driven Architecture

MDE: Model-Driven Enginnering

MILP: Mixed-Integer Linear Programming

MINLP: Mixed-Integer Non Linear Programming

MMI: Man Machine Interface

OCL: Object Constraint Language

OMG: Object Management Group

OO: Object Oriented

PSE: Process System Engineering

RUP: Rational Unified Process

SBVR: Semantic for Business Vocabulary and Rules

SysML: System Modeling Language

UML: Unified Modeling Language

VB: Visual Basic

XML: Extensible Markup Language